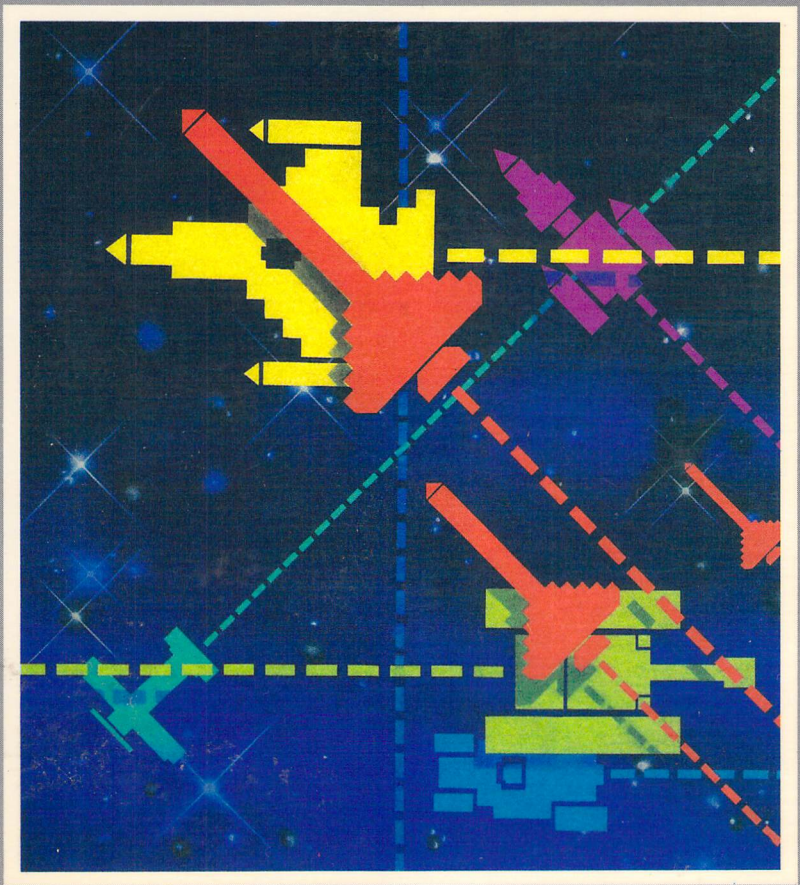
SAMS PRESENTS 22188

VIC 20: 50 Easy-to-Run Computer Games

Edward Burns



VIC 20: 50 Easy-to-Run Computer Games

Congratulations on your decision to use these pre-recorded, fully debugged programs on your VIC 20!

This high-quality cassette is ready to run on your VIC without any modification, and contains every program you see in the book. With it, you'll be able to enjoy many additional hours running programs and modifying them, instead of spending that time manually keying-in each program from the book and looking for any typing errors afterward.

To give you a measure of safety in case of accident, the tape has been recorded with two complete sets of programs, one on each side of the tape. Should the side you're using become unplayable for some reason, just flip the cassette over and you'll have a fresh set of the same programs, ready to go.

HOW TO LOAD THESE CASSETTE PROGRAMS INTO YOUR VIC-20

This tape will load according to the LOAD instructions in your VIC 20 user's manual (the one that came with your computer). If you don't have the manual, here's how to do it:

1. Connect and power up your VIC 20, a TV screen, and the VIC 20 Datasette. Open the door of your Datasette and insert the program cassette in the same way you would an audio cassette. Close the door. Press no buttons yet.
2. Suppose the name of the program you want to load is NIFTY. Using your VIC 20 keyboard, type **LOAD "NIFTY"** just as you see it here, including the space and quotes. Press RETURN.
3. Your VIC will respond on-screen with **PRESS PLAY ON TAPE.** You then depress the "Play" button on the Datasette.
4. The tape starts, your VIC displays **SEARCHING FOR NIFTY,** and the tape continues to roll. As each program passes through, VIC displays **FOUND** followed by the program name. When NIFTY arrives, VIC displays **FOUND NIFTY** and then shows the single word **LOADING.** When the tape stops, the program will have been loaded into RAM and the screen will say **OK READY.** You're ready to RUN and enjoy your program at any time.
5. When you're ready to load another program, repeat the process, using the name of the new program. Your VIC will automatically delete the old program from memory as it loads the new one. The program on the tape, of course, remains undisturbed and can be reused many times.

ALTERNATE WAYS TO LOAD FROM CASSETTE

A. If the program you want to load is one of the last ones on the tape, you can get pretty bored waiting for the others to pass through before yours comes up.

   To avoid this, set the tape counter on your Datasette to zero and then start the loading process. Note the counter reading as each program starts through and write the number in your book.

   If you make it a habit to zero the counter each time you insert a cassette, you can use your Fast Forward button to quickly run the tape ahead until the counter shows the index number for the start of the program you want. Then LOAD it by name as described earlier.

B. A similar—but less accurate—way is to use the index marks molded into the cassette body just under the window between the spindle holes. By noting the index mark closest to the rim of the tape, you can get a rough idea of where each program starts and go from there.

C. If you don't care about time, just insert the cassette, type LOAD with no program name following, and

press RETURN. Your VIC will load the first program on the tape, and then load each succeeding program each time you retype the command. This is a good way to explore the tape as well as to get the tape-counter index number as each program starts.

PROGRAMS ON THIS TAPE, WITH LOCATIONS

Coin Toss	000	Moving Target	182
The 500	012	Keyboard	188
The Great Duck Race	020	Stop	194
Battleship	028	Programmer	199
Fortune-Teller	036	Rock, Scissors, Paper	205
Hurray, Beethoven!	045	Around the Clock	211
Splat	054	Maze	217
Cows and Bulls	061	Shooting Gallery	223
Countdown	069	Magic Square	229
Word Builder	077	Quarterback	234
Golf	083	Math Whiz	241
Copycat	090	Decoding	247
Bird Watching	097	Race Car	252
Blue and Red	103	Sort	257
Gorp Words	110	Fingerquick	263
Mississippi	117	Computer Brain	267
Hole Digging Contest	124	Intercept	272
Diamond	131	Concentration	278
The Backward Test	137	Pickle	284
Nim	143	Line Checkers	289
Darts	150	Birthday	294
Pyramids	157	The Run	299
Risk	164	Gobbler	304
Home Run	170	Flash	310
Fan Tan	176	Tree Climb	313

MORE SAMS VIC 20 BOOKS YOU CAN ENJOY

VIC-20: 50 EASY-TO-RUN COMPUTER GAMES

A collection of games for the unexpanded VIC-20, most of which have 15 to 20 program statements. By Edward Burns.

BOOK ONLY: 96 pages, 5½ x 8½, soft. ©1983.
Ask for No. 22188 ..... $5.95

TAPE CASSETTE OF PROGRAMS ONLY
Ask for No. 22287 ..... $7.95

BOOK PLUS TAPE CASSETTE: Packed in 6 x 9 hardcover vinyl binder with cassette storage feature.
Ask for No. 26170 ..... $12.95

VIC-20 GAMES, GRAPHICS, AND APPLICATIONS

Shows you how to use the VIC-20's user-definable character sets, 4 musical notes, real-time clock, color, and graphics with 24 BASIC programs most of which run on any 5K unexpanded VIC-20. By David Busch.

BOOK ONLY: 136 pages, 5½ x 8½, soft. ©1983.
Ask for No. 22189 ..... $8.95

TAPE CASSETTE OF PROGRAMS ONLY
Ask for No. 22280 ..... $7.95

BOOK PLUS TAPE CASSETTE: Packed in 6 x 9 hardcover vinyl binder with cassette storage feature.
Ask for No. 26167 ..... $15.95

VIC-20 PROGRAMMER'S NOTEBOOK

A collection of VIC-20 programming techniques, hints, kinks, tricks, subroutines, and shortcuts. By Earl R. Savage. 256 pages, 5½ x 8½, comb. ©1983.
Ask for No. 22089 ..... $14.95

VIC-20 PROGRAMMER'S REFERENCE GUIDE

Covers VIC-20 BASIC, machine language, I/O operations, programming tips, advice, and more. By Commodore Computer. 289 pages, 5½ x 8½, comb. ©1982.
Ask for No. 21948 ..... $16.95

These and other Sams Books and Software products are available from better book and software retailers worldwide, or directly from Sams. To order, call 800-428-SAMS or 317-298-5566 and charge your selection to your MasterCard or VISA account.

VIC 20: 50 Easy-to-Run Computer Games

Edward Burns is an Associate Professor in the School of General Studies and Professional Education at the State University of New York at Binghamton. He has written several books and numerous articles relating to education, special education, educational testing, and programming. Mr. Burns has taught in regular and special education, and has taught a computer programming course for gifted fourth, fifth, and sixth grade students.

VIC 20: 50 Easy-to-Run Computer Games

by

Edward Burns

Howard W. Sams & Co., Inc.
4300 WEST 62ND ST. INDIANAPOLIS, INDIANA 46268 USA

Copyright © 1983 by Edward Burns

FIRST EDITION
FIRST PRINTING—1983

All rights reserved. No part of this book shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from the publisher. No patent liability is assumed with respect to the use of the information contained herein. While every precaution has been taken in the preparation of this book, the publisher assumes no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained herein.

International Standard Book Number: 0-672-22188-8
Library of Congress Catalog Card Number: 83-50375

Edited by: *Jim Rounds*

Printed in the United States of America.

PREFACE

The 50 easy-to-run Commodore VIC 20 computer programs in this book offer a variety of games involving skill, chance, and memory. In a very short period of time, you can be competing against the computer in a game of Nim, blasting targets in Shooting Gallery, or testing your attention in a game of Flash.

The games are interesting and challenging, yet each can be entered and run with little difficulty. The programs offer a wide range of entertaining easy-to-run games that do not require many hours of entering statements and debugging. Each program contains less than 30 statements.

The 50 easy-to-run games in this book are designed to entertain, to teach, and to help you explore the many fascinating features of the VIC 20.

Each program is accompanied by a description, a sample run, and a complete program listing. Watch out for bugs and good programming.

EDWARD BURNS

To Andrea and Paul White

CONTENTS

Game		Page
	Programming Hints	9
1.	Coin Toss	13
2.	The 500	15
3.	The Great Duck Race	17
4.	Battleship	19
5.	Fortune-Teller	21
6.	Hurray, Beethoven!	23
7.	Splat	25
8.	Cows and Bulls	29
9.	Countdown	31
10.	Word Builder	33
11.	Golf	35
12.	Copycat	37
13.	Bird Watching	39
14.	Blue and Red	41
15.	Gorp Words	43
16.	Mississippi	45
17.	Hole Digging Contest	47
18.	Diamond	49
19.	The Backward Test	51
20.	Nim	53
21.	Darts	55
22.	Pyramids	59
23.	Risk	61
24.	Home Run	63
25.	Fan Tan	65
26.	Moving Target	67
27.	Keyboard	69
28.	Stop	71
29.	Programmer	73
30.	Rock, Scissors, Paper	75

31. Around the Clock 77
32. Maze 79
33. Shooting Gallery 81
34. Magic Square 83
35. Quarterback 85
36. Math Whiz 89
37. Decoding 93
38. Race Car 95
39. Sort 97
40. Fingerquick 99
41. Computer Brain 101
42. Intercept 103
43. Concentration 105
44. Pickle 109
45. Line Checkers 111
46. Birthday 113
47. The Run 115
48. Gobbler 117
49. Flash 119
50. Tree Climb 121

PROGRAMMING HINTS

These short BASIC programs are for the VIC 20 microcomputer without the expansion board option. If your VIC 20 has the expansion board, you must remove it so these 50 easy-to-run programs will load and run properly.

Each program in this book contains less than 30 statements, with one statement assigned to each line. Although each program can be entered in a very short period of time, bugs can occur that will prevent the program from running. As the saying goes, "If you haven't had a bug, you haven't done much programming."

The following suggestions will help you minimize programming bugs so that you can move on to the task of computer game playing.

1. Copy each program exactly as shown in the program listing. Changing a comma to a period, even though a very small change, can totally prevent a program from running. The more careful you are when entering programs, the more likely you are that a program will run successfully.
2. If you enter a program and encounter a bug, use the information given by the computer to correct the bug. As an example, consider the following program:

```
10 REM SAMPLE
15 A=25
20 PRIMT A
25 END
```

If you run this program, the computer will respond with an error message:

```
?SYNTAX
ERROR IN 20
```

This message indicates that there is a syntax error in statement 20. To correct this error, first display statement 20 so

that you can see what the error is. This is achieved by typing

```
LIST 2Ø
```

and the computer will print statement 20:

```
2Ø PRIMT A
```

The syntax error occurred because the word PRIMT is misspelled. This bug is corrected by reentering statement 20 with the correct spelling:

```
2Ø PRINT A
```

Now, when the program is run, the computer will print the number 25 that is stored in variable A.

3. In order to inspect a program, it is sometimes useful to list more than one statement. An entire program can be listed on the screen by typing

```
LIST
```

A portion of a program can be listed by using the dash with the LIST command. As an example, the following command instructs the computer to list all the statements up to and including statement 50. The statements in the program after statement 50 are not printed. The LIST command is used frequently when debugging, and also to check your work as you are entering programs.

```
LIST 5Ø
```

4. How do you stop a program when it is running? There are two ways: First, simply press the RUN/STOP key on the far left of the keyboard. The computer will respond with something like this:

```
BREAK IN 65
```

This indicates that the program was interrupted while at statement number 65.

If you attempt to interrupt a program at an INPUT statement (or when a question mark prompt appears on the screen), you must press the RUN/STOP key and the RESTORE key (on the far right of the keyboard) at the same time.

5. Here is how to use PRINT statements to clear the screen, skip spaces, and skip lines. When you type a quote after a print statement, hold down the shift key and press the CLR/HOME key, and then type an end quote, the computer will print an inverse heart in quotes. When the computer encounters this statement in a program, it will clear the screen. The following is an example:

```
10 PRINT ''♥''
```

Pressing the up-down cursor (CRSR) key after a quote will cause the computer to print an inverse Q. Each inverse Q will cause the computer to skip a line. Pressing the left-right CRSR key will cause the computer to print a different inverse character. Each of these characters will cause the computer to skip a space.

Take a look at statement 50 in the program called Flash for an example of how the PRINT statement can be used to clear the screen, skip lines, and spaces. In this statement, the inverse heart causes the computer to clear the screen, the eight inverse Q's cause the computer to skip eight lines, and the next inverse character (printed seven times) causes the computer to skip seven spaces.

6. Use your VIC 20 computer guide! As you become familiar with the manual and the VIC 20 BASIC language, your skill as a programmer will develop. There will always be questions as to why something doesn't work, how a program can be changed, or how a program can be improved. For the most part, the answers to these questions can be found in the VIC 20 computer guide. If you have a question, consult your VIC 20 computer guide.

GAME 1

COIN TOSS

Description:

Coin Toss is a game of chance against the computer. A "coin" is tossed by random numbers. The RND function is used to generate either a 0 or a 1. If the RND function results in a 0, this is called a TAIL and one point is given to the computer. If the RND function results in a 1, this is called a HEAD and you are given one point.

To toss a coin, press the RETURN key after the computer prompts with this message:

```
TOSS?
```

The winner is the first to reach 10 points. If you lose to the computer, try again . . . and again . . . and again. Your luck is bound to improve!

Sample Run:

```
RUN
TOSS?
HEADS
YOUR SCORE= 1
COMPUTER= Ø

TOSS?
TAILS
YOUR SCORE= 1
COMPUTER= 1

TOSS?
```

```
TAILS
YOUR SCORE= 1
COMPUTER= 2
.
.
.
.
.
TOSS?
HEADS
YOUR SCORE= 1Ø
COMPUTER= 6
YOU WON!
```

PROGRAM:

```
10 REM COIN TOSS
15 INPUT "TOSS";T$
20 R=INT(RND(1)*2)
25 IF R=1 THEN 45
30 C=C+1
35 PRINT "TAILS"
40 GOTO 55
45 Y=Y+1
50 PRINT "HEADS"
55 PRINT "YOUR SCORE="   Y
60 PRINT "COMPUTER="C
65 IF C=10 THEN 85
70 IF Y=10 THEN 95
75 PRINT
80 GOTO 10
85 PRINT "COMPUTER WON   "
90 GOTO 100
95 PRINT "YOU WON!"
100 END

READY.
```

GAME 2

THE 500

Description:

The 500 is a game of time and patience. The computer will count from 1 to 500. However, you will not see the computer counting. To win The 500 you must stop the computer's counting (by pressing the RETURN key) at or between 480 and 500.

If you stop the computer at or between 480 and 500, YOU ARE A WINNER. If you stop the computer before it reaches 500, the computer will print your LAP TIME. This is the number the computer reached when it was stopped. If you fail to stop the computer before it finished counting, you CRASH.

The computer begins counting after this message is printed:

```
START YOUR ENGINES
```

Remember that in order to stop the computer, you must press the RETURN key.

Sample Run:

```
RUN
START YOUR ENGINES
LAP TIME= 300

START YOUR ENGINES
LAP TIME= 456

START YOUR ENGINES
LAP TIME= 450
```

```
START YOUR ENGINES
CRASH!!!!!

READY.

RUN
START YOUR ENGINES
LAP TIME= 400

START YOUR ENGINES
LAP TIME= 484
YOU ARE A WINNER!!
```

PROGRAM:

```
10 REM THE 500
15 PRINT "START YOUR ENGINES"
20 FOR X=1 TO 500
25 GET A$
30 IF A$<>"" THEN 50
35 NEXT X
40 PRINT "CRASH!!!!!"
45 GOTO 75
50 PRINT "LAP TIME="X
55 IF X>479 THEN 70
60 PRINT
65 GOTO 15
70 PRINT "YOU ARE A WINNER!!"
75 END

READY.
```

GAME 3

THE GREAT DUCK RACE

Description:

The Great Duck Race pits five speedy ducks against one another. The excitement of a thrilling duck race is tremendous!

After the program has been entered, press the RETURN key to start the race. Once the race is started, the names of the speedy little ducks (watch out for Homer) will flash on the screen as they jockey for position.

At the end of the race, the speed for each duck is printed. The duck with the highest speed index is the winner.

Happy duck racing!

Sample Run:

```
RUN
START RACE?
DAF
HOMER
HOMER
FRED
DAF
QUACK
QUACK
QUACK
QUACK
QUACK
.
.
.
```

```
.
.
DAF
QUACK
FRED
HOMER
SQUAWK

DUCK        SPEED
DAF         49
QUACK       43
FRED        55
SQUAWK      45
HOMER       58
```

PROGRAM:

```
10 REM THE GREAT DUCK RACE
15 FOR X=1 TO 5
20 READ D$(X)
25 NEXT X
30 DATA DAF,QUACK,FRED    ,SQUAWK,HOMER
35 INPUT "START RACE";    S$
40 FOR X=1 TO 250
45 R=INT(RND(1)*5+1)
50 PRINT D$(R)
55 T(R)=T(R)+1
60 NEXT X
65 PRINT
70 PRINT "DUCK"TAB(12)    "SPEED"
75 PRINT
80 FOR X=1 TO 5
85 PRINT D$(X),T(X)
90 NEXT X
95 END

READY.
```

GAME 4

BATTLESHIP

Description:

You are a submarine commander after a huge battleship (or it's after you). You have eight torpedoes to sink the enemy ship. If you fail to sink the enemy in ten trys, you sink. Being a submarine commander is no bowl of soup.

You fire a torpedo by entering two coordinates. Each coordinate is a number between 1 and 100. The coordinates you enter show where you think the enemy battleship is.

After two coordinates have been entered, a torpedo is fired. The computer will show you how close you came to hitting the enemy ship by printing a DISTANCE index. If your distance is ever 5 or less, you sink the enemy battleship.

You only have eight torpedoes, so use the DISTANCE index to adjust your coordinates.

Sample Run:

```
RUN
#1 COORDINATE? 5Ø
#2 COORDINATE? 5Ø
DISTANCE= 41

#1 COORDINATE? 75
#2 COORDINATE? 5Ø
DISTANCE? 65

#1 COORDINATE? 25
#2 COORDINATE? 5Ø
```

```
DISTANCE= 18

#1 COORDINATE? 10
#2 COORDINATE? 50
DISTANCE= 11

#1 COORDINATE? 10
#2 COORDINATE? 60
DISTANCE= 21

#1 COORDINATE? 10
#2 COORDINATE? 40
DISTANCE= 1
VICTORY!
```

PROGRAM:

```
10 REM BATTLESHIP
15 A=INT(RND(1)*100+1)
20 B=INT(RND(1)*100+1)
25 PRINT
30 PRINT
35 INPUT "#1 COORDINATE";X
40 INPUT "#2 COORDINATE";Y
45 D=SQR((A-X)^2+(B-Y)^2)
50 D=INT(D)
55 PRINT "DISTANCE="D
60 IF D<6 THEN 80
65 T=T+1
70 IF T=8 THEN 90
75 GOTO 25
80 PRINT "VICTORY!"
85 GOTO 95
90 PRINT "YOU SUNK"
95 END

READY.
```

GAME 5

FORTUNE-TELLER

Description:

Fortune-Teller will predict your future in four areas: the number of children that you will have, the type of house that you will live in, your wealth, and your job.

By changing the DATA statements, you can enter your own list of items. Each DATA statement should contain a house, wealth, and job description, in that order. You can change the prediction for the number of children that you will have by changing the number 20 in statement 70. You might want to do this if the computer predicts that you will have 19 children!

Never fear, if you find that you are a mule skinner living in a cave and you only have $2.47 in the bank . . . simply press the P key for another prediction.

Sample Run:

```
RUN
PREDICTION? P
YOUR FUTURE:
CHILDREN: 7
HOUSE: MANSION
WEALTH: INHERITED $1000000
JOB: MUD WRESTLER

PREDICTION? P
YOUR FUTURE:
CHILDREN: 15
HOUSE: TREE
```

```
WEALTH: TYCOON
JOB: WORM FARM OPERATOR

PREDICTION? P
YOUR FUTURE:
CHILDREN: 18
HOUSE: LOG CABIN
WEALTH: FILTHY RICH
JOB: MULE SKINNER
```

PROGRAM:

```
10 REM FORTUNE-TELLER
15 FOR X=1 TO 5
20 READ H$(X),W$(X),J$(X)
25 NEXT X
30 DATA CASTLE,TYCOON,KING
35 DATA MANSION,FILTHY RICH,FAMOUS ACTOR
40 DATA CAVE,INHERITED $1000000,WORM FARM OPERATOR
45 DATA TREE,$2.47 IN THE BANK,MUD WRESTLER
50 DATA LOG CABIN,OWE $1000000,MULE SKINNER
55 INPUT "PREDICTION";P$
60 IF P$<>"P" THEN 100
65 PRINT "YOUR FUTURE:"
70 PRINT "CHILDREN:"INT(RND(1)*20)
75 PRINT "HOUSE: "H$(INT(RND(1)*5+1))
80 PRINT "WEALTH: "W$(INT(RND(1)*5+1))
85 PRINT "JOB: "J$(INT(RND(1)*5+1))
90 PRINT
95 GOTO 55
100 END

READY.
```

GAME 6

HURRAY, BEETHOVEN!

Description:

This game will teach you several musical notes. The notes used in this game are A, B, C, D, and E. True, there are only five notes used in this game. But, just like you, Beethoven had to begin someplace.

When the computer prints LISTEN on the screen, one of the musical notes will be played. The computer will then give the following prompt:

```
A,B,C,D,OR E?
```

You are to enter the note that was played; that is, enter A, B, C, D, or E. If you are incorrect, the computer will tell you which note was played. If you are correct, the computer will print HURRAY, BEETHOVEN.

The computer will sound 15 notes. For each round, a note is randomly selected and played. Your score (0 to 15) will be printed at the end of the game.

Sample Run:

```
RUN

LISTEN:                          (A note is played.)
A,B,C,D,OR E? E
HURRAY, BEETHOVEN!
LISTEN:                          (A note is played.)
A,B,C,D,OR E? B
THE NOTE WAS D
```

```
LISTEN:                          (A note is played.)
A,B,C,D,OR E? B
THE NOTE WAS C

LISTEN                           (A note is played.)
A,B,C,D,OR E? C
HURRAY, BEETHOVEN!
.
.
.
.
.
LISTEN                           (A note is played.)
A,B,C,D,OR E? E
HURRAY, BEETHOVEN!

SCORE= 10
```

PROGRAM:

```
10 REM HURRAY, BEETHOVEN!
15 FOR X=1 TO 5
20 READ N$(X),N(X)
25 NEXT X
30 DATA A,219,B,223,C,225,D,228,E,231
35 FOR X=1 TO 15
40 PRINT
45 PRINT "LISTEN:"
50 POKE 36878,5
55 R=INT(RND(1)*5+1)
60 POKE 36875,N(R)
65 FOR K=1 TO 500
70 NEXT K
75 POKE 36878,0
80 INPUT "A,B,C,D,OR E";A$
85 IF N$(R)=A$ THEN 100
90 PRINT "THE NOTE WAS "N$(R)
95 GOTO 110
100 PRINT "HURRAY, BEETHOVEN!"
105 S=S+1
110 FOR K=1 TO 1000
115 NEXT K
120 NEXT X
125 PRINT
130 PRINT "SCORE="S
135 END

READY.
```

GAME 7

SPLAT

Description:

There is nothing like the thrill and excitement of a serious game of Splat. But don't play Splat if you are not a serious game player. Also, don't play Splat if you are afraid of being splatted (which is what happens when you lose at Splat).

Splat is a counting game. You just count from 1 to 9 . . . sort of. The computer will print numbers randomly, 1 through 9. The first time you see 1, you must press the RETURN key or you will get splatted. After this is done, you must then press the RETURN key as soon as the next 2 appears. Next, you must press the RETURN key as soon as the next 3 appears. This procedure is repeated all the way through 9.

As Splat players across the land will quickly note, the game of Splat demands a high level of concentration. Pay attention and try not to get you know what!

Sample Run:

```
RUN
 8
 2
 4
 1
   1
 7
 6
 3
```

(The first number in the sequence is 1, so the RETURN key is pressed as soon as the 1 appears.)

```
 6
 1
 2
  2
 8
 2
 4
 7
 3
  3
 4
  4
 1
 9
 5
SPLAT SPLAT SPLAT SPLA
T SPLAT SPLAT SPLAT SP
LAT SPLAT SPLAT SPLAT
SPLAT SPLAT SPLAT SPLA
T SPLAT SPLAT SPLAT SP
LAT SPLAT SPLAT SPLAT
SPLAT SPLAT SPLAT SPLA
.
.
.
```

(The next number in the sequence is 2, so the RETURN key is pressed.)

(The next number is 5, but the 5 key is not pressed and . . .)

PROGRAM:

```
10 REM SPLAT
15 FOR X=1 TO 9
20 R=INT(RND(1)*9)+49
25 PRINT TAB(7)CHR$(R)
30 PRINT
35 FOR K=1 TO 100
40 GET A$
45 IF A$<>"" THEN 65
50 NEXT K
55 IF R-48=X THEN 90
60 GOTO 20
65 IF R-48<>X THEN 90
70 PRINT TAB(10);X
75 NEXT X
80 PRINT "SAVED FROM SPLAT"
85 GOTO 105
90 FOR K=1 TO 150
95 PRINT "SPLAT ";
100 NEXT K
105 END

READY.
```

GAME 8

COWS AND BULLS

Description:

The object of this very old game of logic is to guess three digits. Each digit is a number between 1 and 9. Clues are given by indicating the number of "cows" and "bulls." One bull means that a number in the correct position was correctly guessed; one cow means that a number was guessed, but the number is not in the correct position. Three bulls mean that you have correctly guessed the three numbers in the correct positions. Three cows mean that you have correctly guessed the three numbers but that the positions of the numbers are incorrect.

When you have finally guessed the three numbers in the correct positions (and have 3 BULLS), the number of guesses you made will be printed.

To make the task even more difficult, delete statement 10 so that duplicate numbers can occur.

Sample Run:

```
RUN
THREE NUMBERS? 1,2,3
COWS= Ø     BULLS= Ø

THREE NUMBERS? 4,5,6
COWS= 2     BULLS= Ø

THREE NUMBERS? 1,5,6
COWS= 2     BULLS= Ø
```

```
THREE NUMBERS? 1,6,5
COWS= 1     BULLS= 1

THREE NUMBERS? 6,1,5
COWS= 2     BULLS= Ø

THREE NUMBERS? 5,6,1
COWS= Ø     BULLS= 2

THREE NUMBERS? 5,6,7
COWS= Ø     BULLS= 2

THREE NUMBERS? 5,6,8
COWS= Ø     BULLS= 3

GUESSES= 8
```

PROGRAM:

```
10 REM COWS AND BULLS
15 X=INT(RND(1)*9+1)
20 Y=INT(RND(1)*9+1)
25 Z=INT(RND(1)*9+1)
30 IF X=Y OR Y=Z OR X=Z THEN 10
35 BU=0
40 CO=0
45 INPUT "THREE NUMBERS";A,B,C
50 G=G+1
55 IF A=X THEN BU=BU+1
60 IF B=Y THEN BU=BU+1
65 IF C=Z THEN BU=BU+1
70 IF A=Y OR A=Z THEN CO=CO+1
75 IF B=X OR B=Z THEN CO=CO+1
80 IF C=X OR C=Y THEN CO=CO+1
85 PRINT "COWS="CO,"BULLS="BU
90 PRINT
95 IF BU=3 THEN 105
100 GOTO 35
105 PRINT "GUESSES="G
110 END

READY.
```

GAME 9

COUNTDOWN

Description:

You must have good eye-hand coordination to be an Olympic caliber Countdown player. The computer counts down from 50 to 0. After the computer reaches 0, you must press the RETURN key (or space bar or any other key) to stop the computer from printing hearts.

If you press a key before the computer reaches 0, you will receive a score of ZIP. A ZIP is something like being zapped or splatted, and means that you receive a score of 0. A perfect score is 60 (which I have never seen); a good score is between 48 to 52. If you cannot receive a Countdown score of 25 or higher, Countdown is obviously not your game.

Sample Run:

```
5Ø
49
48
47
.
.
.
.
.
5
4
3
```

```
2

1
Ø
♥
♥
♥
♥
♥
♥
♥
♥
♥
♥
♥
```

(If the RETURN key is pressed before 0, you get zipped.)

(The RETURN key is pressed at this point.)

```
SCORE= 49
```

PROGRAM:

```
10 REM COUNTDOWN
15 PRINT "BEGIN COUNTDOWN"
20 FOR X=50 TO 0 STEP -1
25 FOR K=1 TO 100
30 NEXT K
35 PRINT TAB(9);X
40 GET A$
45 IF A$<>"" THEN 90
50 NEXT X
55 FOR X=60 TO 1 STEP -1
60 PRINT TAB(10);CHR$(115)
65 GET A$
70 IF A$<>"" THEN 80
75 NEXT X
80 PRINT "SCORE="X
85 GOTO 95
90 PRINT "SCORE= ZIP"
95 END

READY.
```

GAME 10

WORD BUILDER

Description:

This program will generate 10 letters. The task is to use as many of the letters as possible to make a word. Use the following table to assign points for every letter generated by the computer that is used in a word. After you use the letters to spell a word, determine your point value.

Letters	Points
E T O A N I	1
R S H D L	2
C W U M F	3
Y G P B V	4
K X Q J Z	5

Press the RETURN key for letters and type Y if you want the program to continue.

Sample Run:

```
RUN
PRESS RETURN?
 T
 O
 A
 I
 F
 Y
 L
 Z
```

```
 B
 H
CONTINUE?
PRESS RETURN?
 X
 R
 U
 R
 O
 Q
 C
 L
 B
 L
CONTINUE?
```

PROGRAM:

```
10 REM WORD BUILDER
15 PRINT
20 INPUT "PRESS RETURN";R$
25 FOR X=1 TO 10
30 PRINT TAB(3);CHR$(INT(RND(1)*26+65))
35 PRINT
40 NEXT X
45 INPUT "CONTINUE";C$
50 IF C$="Y" THEN 15
55 END

READY.
```

GAME 11

GOLF

Description:

There is nothing like a relaxing game of computer golf. The game of Golf consists of nine holes. Of course this can easily be changed to 18 holes by changing statement 15 to

```
15 FOR X=1 TO 18
```

Before each shot, the computer will print the hole you are playing (1,2,3, etc.) and the length of the hole in yards. After each shot, the computer will print the yards remaining before you reach the hole.

You must select a club for each shot. The computer will prompt with CLUB and you must enter a number between 1 and 9. Which club should you use? The lower the club number, the farther the ball will go. A 1 club will send the ball down the computer fairway anywhere from 99 to 198 yards, while a 9 club will send the ball between 11 to 22 yards. After a little experimentation, you'll know which club to use (more or less).

By the way, if you hit the ball more than 15 yards beyond the hole, you will be penalized. Golf is a game filled with hazards.

Sample Run:

```
RUN
HOLE 1 YARDS 325
CLUB? 1
YARDS TO GO 154
CLUB? 7
YARDS TO GO 1Ø3
CLUB? 5
```

```
HOLE 1 SCORE 3
TOTAL SHOTS 3

HOLE 2 YARDS 323
CLUB? 1
YARDS TO GO 125
CLUB? 4
HOLE 2 SCORE 2
TOTAL SHOTS 5
.
.
.
.
.
HOLE 9 YARDS 198
CLUB? 1
YARDS TO GO 18
CLUB? 9
HOLE 9 SCORE 2
TOTAL SHOTS 35
```

PROGRAM:

```
10 REM GOLF
15 FOR X=1 TO 9
20 P=0
25 S=INT(RND(1)*400+149)
30 PRINT "HOLE"X"YARDS"S
35 INPUT "CLUB";C
40 S=S-(10-C)*INT(RND(1)*12+11)
45 P=P+1
50 IF S<-15 THEN 70
55 IF S<15 THEN 75
60 PRINT "YARDS TO GO"S
65 GOTO 35
70 P=P+4
75 PRINT "HOLE"X"SCORE"P
80 B=B+P
85 PRINT "TOTAL SHOTS"B
90 PRINT
95 NEXT X
100 END

READY.
```

GAME 12

COPYCAT

Description:

How fast can you copy a series of letters? The computer will print ten letters. As soon as you copy all of the letters, exactly as shown, the computer will indicate your copying speed.

Little is known about the Copycat world record, but some say that it is about 3.5 seconds. However, a time of less than 15 seconds is good, and a time of less than 10 seconds is very good among Copycat players.

If you copy the letters incorrectly, the computer will print that famous Copycat message NO COPY, CAT.

Sample Run:

```
RUN

M P O U F J K E E V
? MPOUGJKEEV          (Do not put spaces between
NO COPY, CAT          letters when entering your
                      answer.)
RUN
C W P F N E L Q D A
? CWPFNELQDA
TIME= 9.Ø5
```

PROGRAM:

```
10 REM COPYCAT
15 PRINT "(CLR)"
20 PRINT
25 FOR X=1 TO 10
30 A$(X)=CHR$(INT(RND(1)*26+65))
35 NEXT X
40 S=TI
45 FOR X=1 TO 10
50 PRINT " "A$(X);
55 A$=A$+A$(X)
60 NEXT X
65 PRINT
70 INPUT T$
75 F=TI
80 IF T$=A$ THEN 95
85 PRINT " NO COPY, CAT"
90 GOTO 100
95 PRINT "TIME="(F-S)/60
100 END

READY.
```

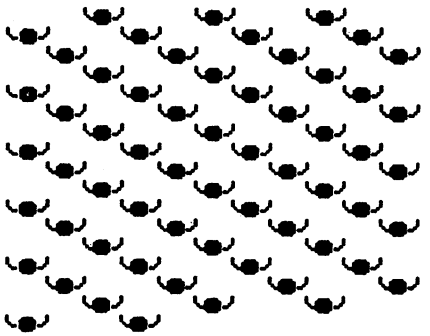
GAME 13

BIRD WATCHING

Description:

The computer will print 10 to 109 "birds." The birds will be shown on the screen for only a few moments, then the screen will be cleared. Once the screen is cleared, you must estimate the number of birds that appeared. A score of ten or less qualifies you as a certified bird watcher, and a score of five, or less, rates you as a bird-watching expert.

Sample Run:



PROGRAM:

```
10 REM BIRD WATCHING
15 R=INT(RND(1)*100+10)
20 B$="JQK"
25 PRINT "(CLR)"
30 FOR X=1 TO R
35 PRINT " "B$;
40 NEXT X
45 FOR X=1 TO 1000
50 NEXT X
55 PRINT "(CLR)"
60 INPUT "BIRDS";B
65 PRINT
70 PRINT "SCORE="B-R
75 END

READY.
```

GAME 14

BLUE AND RED

Description:

So you think you know your colors? So you think it's easy to read simple words like RED and BLUE? You might sing a different tune when you play Blue and Red.

You need a color tv to play this game. The computer will print either the word BLUE or RED on the screen. Whenever the computer prints the word BLUE, you must press the RETURN key; otherwise you will get an OOPS! and the game will be over. You are to do nothing when the word RED appears.

All this seems very simple, except for the fact that the word RED is sometimes red in color and sometimes blue; while the word BLUE is sometimes blue in color and sometimes red. Statements 45 and 50 determine what color will be used. In statement 45, the strange character in the parentheses is obtained by pressing the CTRL (control) and the 7 key (BLU is printed on the front of the key) at the same time. In statement 50, the character in parentheses is obtained by pressing the CTRL key and the 3 key (RED is printed on the front of the key) at the same time.

Sample Run:

```
RUN
RED <---- (The RETURN key is not pressed even
BLUE <--- though RED is printed in the color blue.)
RED       (The RETURN key is pressed even though
RED       the word BLUE is printed in the color
BLUE      red.)
```

```
 RED
 BLUE
 RED
OOPS!
RUN
 RED
 BLUE
 RED
 BLUE
 BLUE
 .
 .
 .
 .
 .
 RED
 BLUE
GREAT!
```

PROGRAM:

```
10 REM BLUE AND RED
15 C$(0)="BLUE"
20 C$(1)="RED"
25 PRINT
30 PRINT
35 FOR X=1 TO 25
40 R=INT(RND(1)*2)
45 IF R=0 THEN PRINT "(BLU)"
50 IF R=1 THEN PRINT "(RED)"
55 PRINT
65 R=INT(RND(1)*2)
70 PRINT TAB(5);C$(R)
75 FOR K=1 TO 300
80 GET T$
85 IF T$<>"" THEN 110
90 NEXT K
95 IF C$(R)="BLUE" THEN 130
100 GOTO 115
110 IF C$(R)="RED" THEN 130
115 NEXT X
120 PRINT "GREAT!"
125 GOTO 135
130 PRINT "OOPS!"
135 END

READY.
```

GAME 15

GORP WORDS

Description:

A *gorp* word is a string of letters randomly put together. The following are examples of gorp words: SDX, LOPU, RTOPCERMMK, HTFASCXFGOPINMFQY.

In the game of Gorp Words, the task is to remember the longest word possible. The game begins with a single letter. The letter will flash on the screen for a moment, and then you must enter the letter that was shown. Next, two letters will appear, and then three letters, and so on.

The game ends when you fail to enter a string of letters exactly as it was shown. Your score is the number of letters of the longest string you were able to correctly reproduce.

A good gorp score is 6 or better; a gorp score of more than 10 will put you in the Gorp Hall of Fame; and a gorp score of 20 or more (impossible!) means that you are one of the greatest gorp players of all time.

Sample Run:

```
RUN
 W                (letter appears on the screen for a moment)
? W
 DA
? DA
 HZX
? HZX
 QXNY
? QXNY
```

```
 VVAMF
? VVAMF
 DENSQQ
? DENSQQ
 RLINTCZ
? RLINTCZ
 MKLLOPCE
? MKLLOCPE
SCORE= 7
```

PROGRAM:

```
10 REM GORP WORDS
15 FOR X=1 TO 25
20 PRINT CHR$(147)
25 PRINT
30 N=N+1
35 A$=""
40 FOR K=1 TO N
45 A$=A$+CHR$(INT(RND(1)*26+65))
50 NEXT K
55 PRINT TAB(3);A$
60 FOR K=1 TO 2000
65 NEXT K
70 PRINT CHR$(147)
75 PRINT
80 INPUT T$
85 IF T$<>A$ THEN 95
90 NEXT X
95 PRINT "SCORE="N-1
100 END

READY.
```

GAME 16

MISSISSIPPI

Description:

Mississippi is a game of chance and skill. You must outwit (and outluck) the computer. You and the computer are first given a number between 1 and 100. Although the computer does the dealing, it (he, she, whatever) is extremely honest. The goal is to reach a score as close as possible to 100 without going over 100.

You can increase your score by typing Y after the computer prompts MORE. If you are satisfied with your score, type N.

If you go over 100, you lose. If your score is not over 100, the computer will then show its score. The computer will then add to its score until it reaches a score greater than 75 or until it goes over 100. If the computer goes over 100, you win.

If the computer does not go over 100, its score will be somewhere between 76 and 100. If the computer has a final score equal to or greater than your score, you lose. If your final score is greater than the computer's score, you win.

You play against the computer 10 times. You receive 100 credits for each win, and lose 100 for each loss. At the end of each game the computer will show your total CREDITS. If your CREDITS is a minus value (e.g., −200), you owe the computer.

Sample Run

```
RUN
YOU HAVE 6Ø
MORE? Y
YOU HAVE 1Ø1
YOU LOSE
```

```
CREDITS=-100

YOU HAVE 49
MORE? Y
YOU HAVE 84
MORE? N
COMPUTER HAS 54
COMPUTER HAS 73
COMPUTER HAS 119
YOU WIN!
CREDITS= 0
.
.
.
.
.
```

PROGRAM:

```
10 REM MISSISSIPPI
15 FOR X=1 TO 10
20 C=INT(RND(1)*100+1)
25 Y=INT(RND(1)*100+1)
30 PRINT
35 PRINT "YOU HAVE"Y
40 IF Y>100 THEN GOTO 120
45 INPUT "MORE";A$
50 IF A$<>"Y" THEN 65
55 Y=Y+INT(RND(1)*50+1)
60 GOTO 35
65 PRINT "COMPUTER HAS"C
70 IF C>75 THEN 90
75 C=C+INT(RND(1)*50+1)
80 IF C>100 THEN 105
85 GOTO 65
90 IF C>=Y THEN 120
95 GOTO 105
100 PRINT "COMPUTER HAS"C
105 PRINT "YOU WIN!"
110 S=S+100
115 GOTO 130
120 PRINT "YOU LOSE"
125 S=S-100
130 PRINT "CREDITS="S
135 NEXT X
140 END

READY.
```

GAME 17

HOLE DIGGING CONTEST

Description:

There is no athletic competition quite like the famous Hole Digging Contest. You begin by entering the number of contestants you wish to participate in this marvelous contest. This can be from 2 to 9 contestants.

Once you enter the number of contestants and press the RETURN key, the race is underway. The computer will show the progress of the hole diggers as the race proceeds. The contestant who has dug the most holes, as one would expect, is the hole digging contest winner. If a tie occurs, the contest must be started over again.

Sample Run:

```
RUN
CONSTANTS? 5
```

(The progress of the hole diggers is charted by the computer until the contest is over.)

PROGRAM:

```
10 REM HOLE DIGGING CONTEST
15 INPUT "CONTESTANTS";C
20 FOR X=1 TO 10*C
25 PRINT "(CLR)"
30 PRINT
35 PRINT
40 FOR K=1 TO 250
45 NEXT K
50 R=INT(RND(1)*C+1)
55 A$(R)=A$(R)+CHR$(209)
60 FOR K=1 TO C
65 PRINT K;A$(K)
70 NEXT K
75 FOR L=1 TO 750
80 NEXT L
85 NEXT X
90 END

READY.
```

GAME 18

DIAMOND

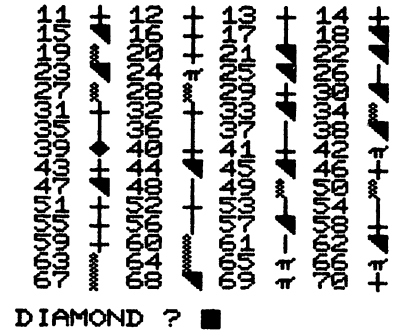
Description:

You can't be careless when you play Diamond. The game begins when the screen fills with various graphic characters, and one diamond. Your task is to find the diamond as soon as possible. When you find the diamond, note the number next to the diamond, and then enter that number after the computer prompt DIAMOND.

The object is to search for the diamond, and to find it as soon as possible. If you enter the wrong number, you will be LOST IN THE MINE. And, of course, the longer your diamond search is, the greater will be your time.

What is a good Diamond search time? A time of less than 4.5 seconds is good, and a time less than 3.0 is excellent.

Sample Run:



PROGRAM:

```
10 REM DIAMOND
15 DIM G$(70)
20 FOR X=11 TO 70
25 G$(X)=CHR$(INT(RND(1)*5+123))
30 NEXT X
35 R=INT(RND(1)*68+11)
40 G$(R)=CHR$(122)
45 C=11
50 F=TI
55 FOR X=1 TO 15
60 FOR K=1 TO 4
65 PRINT C;G$(C);
70 C=C+1
75 NEXT K
80 PRINT
85 NEXT X
90 INPUT "DIAMOND";A
95 S=TI
100 IF A=R THEN 115
105 PRINT "LOST IN THE MINE"
110 GOTO 120
115 PRINT "TIME="(S-F)/60
120 END

READY.
```

GAME 19

THE BACKWARD TEST

Description:

Have you ever taken a multiple-choice test and found yourself guessing? The Backward Test is a multiple-choice test, but the only way to answer the questions is by guessing. There are 20 questions. You are given the question number, and the computer will then prompt with ANSWER. The answer to each question will be 1, 2, 3, or 4.

A perfect score on the Backward Test is 100. There is always the possibility that you will receive a high score (which is a score 50 or greater), but the average score of the Backward Test is about 25.

Sample Run:

```
RUN
QUESTION 1
ANSWER? 4
THE ANSWER WAS 3

QUESTION 2
ANSWER? 1
THE ANSWER WAS 2

QUESTION 3
ANSWER? 4
CORRECT!
.
.
.
```

```
.
.
QUESTION 20
ANSWER?
THE ANSWER WAS 3
BACKWARD SCORE= 15
```

PROGRAM:

```
10 REM THE BACKWARD TEST
15 PRINT "(CLR)"
20 FOR X=1 TO 20
25 R=INT(RND(1)*4+1)
30 PRINT "QUESTION"X
35 INPUT "ANSWER";A
40 IF A=R THEN 55
45 PRINT "THE ANSWER WAS"R
50 GOTO 65
55 PRINT "CORRECT!"
60 S=S+1
65 PRINT
70 PRINT
75 NEXT X
80 PRINT "BACKWARD SCORE="S*5
85 END

READY.
```

GAME 20

NIM

Description:

The computer is an excellent Nim player. The game begins with NIM equal to 15. A "flip" of the old computer coin will decide whether you or the computer goes first.

You play Nim by taking away 1, 2, or 3 points from the current points in NIM. The first one to reach 1 is the winner. As an example, if NIM is equal to 4 and you take away 3, the value left in NIM is 1 and you are the winner.

The white Q on a black background in statement 25 is a cursor control and is used to skip three lines after the initial value in NIM (which is 15) is printed. This symbol is typed by first typing a parenthesis and then pressing the up-down cursor key, or the key marked CRSR with the arrows pointing up and down. Since three lines are to be skipped, press this key three times and then add an ending parenthesis.

Sample Run:

```
RUN
NIM= 15

YOUR TURN? 1
NIM IS NOW 14
C CHANGED NIM TO 12

YOUR TURN? 3
NIM IS NOW 9
C CHANGED NIM TO 8
```

```
YOUR TURN? 2
NIM IS NOW 6
C CHANGED NIM TO 5

YOUR TURN? 2
NIM IS NOW 3
C CHANGED NIM TO 1

COMPUTER WON
```

PROGRAM:

```
10 REM NIM
15 N=15
20 PRINT "(CLR)"
25 PRINT "NIM="N"(DN)(DN)(DN)"
30 IF INT(RND(1)*2+1)=2 THEN 65
35 INPUT "YOUR TURN";Y
40 IF Y<1 OR Y>3 THEN 35
45 IF N-Y<1 THEN 85
50 N=N-Y
55 PRINT "NIM IS NOW"N
60 IF N=1 THEN 120
65 C=INT(RND(1)*3+1)
70 IF N=2 OR N=6 THEN C=1
75 IF N=3 OR N=7 THEN C=2
80 IF N=4 OR N=8 THEN C=3
85 N=N-C
90 PRINT "C CHANGED NIM TO"N
95 PRINT
100 IF N=1 THEN 110
105 GOTO 35
110 PRINT "COMPUTER WON"
115 GOTO 125
120 PRINT "YOU WON!"
125 END

READY.
```

GAME 21

DARTS

Description:

Darts is a game of chance against the computer. You begin each game with 301 points. The goal is to reach 0 first. Each turn consists of three dart throws. You throw a dart by simply pressing the RETURN key. After each throw, between 1 and 20 points will be subtracted from your score. The number of points remaining, for both you and the computer, will be printed after each throw.

Special rules: points will not be subtracted until a dart worth 18, 19, or 20 points is first thrown. If 301 is printed after a number of throws, this means that an initial dart worth 18, 19, or 20 points has not been thrown. Also, to win you must throw a dart that will bring you (or the computer) to 0. If your score is 4, a dart worth 5 points will not count.

Sample Run:

```
RUN
DART?                              (press RETURN key)
YOUR SCORE= 301
DART?
YOUR SCORE= 301
DART?
YOUR SCORE? 301
COMPUTER= 301
COMPUTER= 301
COMPUTER= 281

DART?
```

```
YOUR SCORE= 282
DART?
YOUR SCORE= 273
DART?
YOUR SCORE= 257
COMPUTER= 279
COMPUTER= 266
COMPUTER= 253
.
.
.
.
.
DART?
YOUR SCORE= 1
DART?
YOUR SCORE= Ø

YOU WON!
```

PROGRAM:

```
10 REM DARTS
15 C=301
20 Y=301
25 IF INT(RND(1)*2+1)=2 THEN 70
30 FOR X=1 TO 3
35 R=INT(RND(1)*20+1)
40 IF C=301 AND R<18 THEN 55
45 IF C-R<0 THEN 55
50 C=C-R
55 PRINT "COMPUTER="C
60 IF C=0 THEN 125
65 NEXT X
70 PRINT
75 FOR X=1 TO 3
80 INPUT "DART";D$
85 R=INT(RND(1)*20+1)
90 IF Y=301 AND R<18 THEN 105
95 IF Y-R<0 THEN 105
100 Y=Y-R
105 PRINT "YOUR SCORE="Y
110 IF Y=0 THEN 140
115 NEXT X
120 GOTO 30
125 PRINT
130 PRINT "YOU LOST"
135 GOTO 150
140 PRINT
145 PRINT "YOU WON!"
150 END

READY.
```

GAME 22

PYRAMIDS

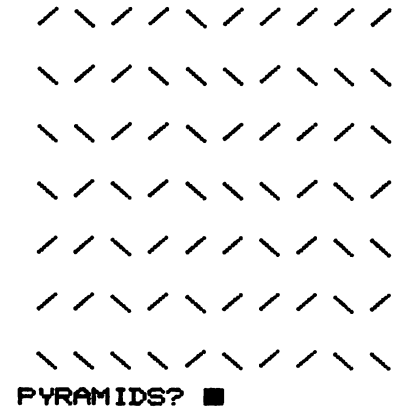
Description:

Two graphic characters are printed in Pyramids: \ and /. Your task is to count the number of times these characters appear in this order: / \. Looking at the direction of the characters, you can see where the game of Pyramids got its name.

If you correctly count the number of Pyramids, you will receive a time. A time of less than 25 seconds is good, less than 20 very good, less than 15 fantastic, and a time of less than 10 seconds makes you an incredible Pyramid player.

One small thing that you should be aware of: if a line ends with a \ and the next line begins with a /, this is counted as a Pyramid. In Pyramid game circles, this is called the old end-of-the-line ploy that is used by the computer to foil Pyramid game players.

Sample Run:



PROGRAM:

```
10 REM PYRAMIDS
15 A$(1)=CHR$(110)
20 A$(2)=CHR$(109)
25 FOR X=1 TO 7
30 FOR K=1 TO 5
35 T(1)=TI
40 C$=A$(INT(RND(1)*2+1))
45 D$=A$(INT(RND(1)*2+1))
50 PRINT " "C$" "D$;
55 IF E$=A$(1) AND C$=A$(2) THEN S=S+1
60 E$=D$
65 IF C$=A$(1) AND D$=A$(2) THEN S=S+1
70 NEXT K
75 PRINT "(DN)(DN)"
80 NEXT X
85 PRINT
90 INPUT "PYRAMIDS";P
95 T(2)=TI
100 IF P=S THEN 110
105 GOTO 120
110 PRINT "TIME="(T(2)-T(1))/60
115 GOTO 125
120 PRINT "PYRAMID MISCOUNT!!"
125 END

READY.
```

GAME 23

RISK

Description:

You begin playing Risk with 100 points. There are 10 plays, and for each play the computer will print a HIGH and LOW number. You than can risk all or a part of the points you have that the next number the computer generates (the X number) will be between the HIGH and LOW numbers.

You cannot risk more points than you have, and the game is over if your point value is less than 1. The object is to accumulate as many points as possible in the 10 plays you are given. If you risk 0 points, you will neither win nor lose points, but you do lose one play.

Sample Run:

```
RUN
HIGH= 1Ø LOW= 6
RISK? 5
X= 1
YOU LOSE
POINTS= 95

HIGH= 5 LOW= 2
RISK? 1
X= 15
YOU LOSE
POINTS= 94

HIGH= 9 LOW= Ø
RISK? 5Ø
```

```
X= 6
YOU WIN!
POINTS= 144
.
.
.
.
.
HIGH= 12 LOW= 11
RISK? Ø
X= 3
YOU LOSE
POINTS= 14
GAME OVER
```

PROGRAM:

```
10 REM RISK
15 P=100
20 H=INT(RND(1)*16)
25 L=INT(RND(1)*H)
30 PRINT
35 PRINT "HIGH="H"LOW="L
40 INPUT "RISK";R
45 IF R>P THEN 40
50 K=INT(RND(1)*16)
55 PRINT "X="K
60 IF K>L AND K<H THEN 80
65 P=P-R
70 PRINT "YOU LOSE"
75 GOTO 90
80 P=P+R
85 PRINT "YOU WIN!"
90 PRINT "POINTS="P
95 IF P<1 THEN 115
100 N=N+1
105 IF N=10 THEN 115
110 GOTO 20
115 PRINT "GAME OVER"
120 END

READY.
```

GAME 24

HOME RUN

Description:

You are given three outs to hit as many home runs as possible. You "swing" the bat by pressing the RETURN key. You will either hit a HOME RUN or make a FLY OUT.

Here's how to interpret the number of home runs you hit:

HOME RUNS	MEANING
15 or more	Hall of Fame
10 to 14	ALL-star
6 to 9	Professional
3 to 5	Minor Leaguer
0 to 2	Try tennis

Sample Run:

```
RUN
SWING?                          (press RETURN key)
HOME RUN!
TOTAL= 1

SWING?
HOME RUN!
TOTAL= 2

SWING?
FLY OUT

SWING?
```

```
HOME RUN!
TOTAL= 3

SWING?
FLY OUT

SWING?
HOME RUN
TOTAL= 4

SWING?
HOME RUN!
TOTAL= 5

SWING?
HOME RUN
TOTAL= 6

SWING?
FLY OUT

NO MORE BATS
```

PROGRAM:

```
10 REM HOME RUN
15 B=INT(RND(1)*350+161)
20 INPUT "SWING";S$
25 IF B>320 THEN 55
30 PRINT "FLY OUT"
35 PRINT
40 K=K+1
45 IF K=3 THEN 80
50 GOTO 15
55 PRINT "HOME RUN!"
60 H=H+1
65 PRINT "TOTAL="H
70 PRINT
75 GOTO 15
80 PRINT "NO MORE BATS"
85 END

READY.
```

GAME 25

FAN TAN

Description:

You begin playing Fan Tan with 100 points. For each play, you select either 1, 2, 3, or 4. The computer then selects a number between 11 and 30, and then subtracts 4 from this number until a number of 4, or less, is reached. If the number you selected is equal to the number finally reached by the computer, you win 40 points. If you did not select the number finally reached by the computer, you lose 10 points.

The game is over after 10 plays, or if your total point value is less than 1 point.

Sample Run:

```
RUN
1,2,3, or 4? 4 (The number selected is 4, and
 22              the number finally reached by the
 18              computer is 2. As a result, 10
 14              points are subtracted from the
 1Ø              score.)
 6
 2
TOTAL POINTS= 9Ø

1,2,3, or 4?
```

PROGRAM:

```
10 REM FAN TAN
15 M=100
20 PRINT
25 INPUT "1,2,3 OR 4";N
30 F=INT(RND(1)*20+11)
35 PRINT TAB(2);F
40 IF F<5 THEN 65
45 F=F-4
50 FOR K=1 TO 750
55 NEXT K
60 GOTO 35
65 IF N=F THEN 80
70 M=M-10
75 GOTO 85
80 M=M+40
85 PRINT "TOTAL POINTS="M
90 C=C+1
95 IF C=10 THEN 110
100 IF M<1 THEN 110
105 GOTO 20
110 PRINT "GAME OVER"
115 END

READY.
```

GAME 26

MOVING TARGET

Description:

The target in this game will move across the screen from left to right. When the target is in your sights, press the RETURN key to fire. If you hit the target (be sure your volume is turned up), you will hear a short beep sound. A perfect score is 20 . . . but you better be a good shot to get a perfect score.

This game is filled with pokes. What do they all mean? In statement 40, POKE 7888,91 positions the sights in the center of the screen. In statement 45, POKE 38608,0 colors the sights black. The pokes in statements 50 and 55 supply the target that moves across the screen. The sound that you hear when you hit the target is controlled by the poke in statement 95 or POKE 36876,240. The volume for this sound is set in statement 15 by POKE 36878,10. Finally, the sound after a hit is ended by the poke in statement 110 or POKE 36876,0. The game of Moving Target illustrates how you can poke around to create a rather interesting game.

Sample Run:

```
RUN
●          +
SCORE=  12
```

(When the target is in the sights, press the RETURN key to fire.)

PROGRAM:

```
10 REM MOVING TARGET
15 POKE 36878,10
20 FOR X=1 TO 20
25 PRINT "(CLR)"
30 PRINT "(DN)(DN)(DN)(DN)(DN)(DN)(DN)"
35 FOR K=1 TO 20
40 POKE 7888,91
45 POKE 38608,0
50 POKE 7877+K,81
55 POKE 38597+K,0
60 FOR L=1 TO 5
65 GET A$
70 IF A$<>"" AND K=11 THEN 95
75 NEXT L
80 POKE 38597+K,1
85 NEXT K
90 GOTO 120
95 POKE 36876,240
100 FOR L=1 TO 100
105 NEXT L
110 POKE 36876,0
115 S=S+1
120 FOR K=1 TO 1000
125 NEXT K
130 NEXT X
135 PRINT "SCORE="S
140 END

READY.
```

GAME 27

KEYBOARD

Description:

Keyboard is a game to determine how well you know the keys on the VIC 20. A character representing one of the keys will be flashed on the screen. This character can be a number, letter, punctuation mark, or special symbol such as a $. The character will remain on the screen until you press the key which corresponds to the letter that is being flashed on the screen.

In all, 30 characters will be presented. The time that you take to complete the task is printed at the end of the game. A time of 60 seconds or less is good; less than 50 seconds is very good; less than 40 seconds means that you've been doing a lot of work on your VIC 20; a time of less than 30 seconds means that you are an expert Keyboard player; and a time of less than 20 seconds means that you are a professional typist.

Sample Run:

```
RUN
 K                    (The character will remain until
                       the corresponding key is pressed.)
 $

 1

 ,

 P
```

```
M

%
.
.
.
.
.
J
TIME= 43.3833333
```

PROGRAM:

```
10 REM KEYBOARD
15 S=TI
20 FOR X=1 TO 30
25 PRINT "(CLR)"
30 PRINT "(DN)(DN)(DN)(DN)(DN)(DN)(DN)"
35 L$=CHR$(INT(RND(1)*58+33))
40 PRINT TAB(10);L$
45 GET A$
50 IF A$=L$ THEN 60
55 GOTO 45
60 NEXT X
65 F=TI
70 PRINT
75 PRINT "TIME="(F-S)/60
80 END

READY.
```

GAME 28

STOP

Description:

Turn up the volume on your tv set and enjoy a friendly game of Stop. There are 15 rounds in the game of Stop. For each round, you will hear a sound that rapidly increases in pitch. By pressing the RETURN key, you can stop the sound and receive a score. If you stop the sound at the very beginning, you will receive a score of 1. If you stop the sound at the very end, you will receive a score of 66. The object is to receive the highest score possible for each round. The computer will print your score for each round and your cumulative score for all 15 rounds.

I almost forgot two additional points of interest. If you fail to press the RETURN key, you will lose 100 points. If you press the RETURN key after the sound, you will lose 100 points and receive only 1 point for the next round.

Here's how to interpret your score:

Score	Meaning
925+	Fantastic
900 - 949	Excellent
850 - 899	Good
750 - 849	OK
500 - 749	Practice

Sample Run:

```
RUN
SCORE 20      TOTAL 20      (The RETURN key
SCORE 38      TOTAL 58      is pressed and a
SCORE 44      TOTAL 102     score of 20 given.)
```

```
SCORE  54     TOTAL 156
SCORE-100     TOTAL  56   (The RETURN key
SCORE  40     TOTAL  96   is not pressed.)
SCORE  62     TOTAL 158
SCORE-100     TOTAL  58   (The RETURN key
SCORE  1      TOTAL  59   is pressed after
SCORE  65     TOTAL 124 the sound ends.)
SCORE  55     TOTAL 179
SCORE  58     TOTAL 237
SCORE  61     TOTAL 298
SCORE  65     TOTAL 363
SCORE  63     TOTAL 426
```

PROGRAM:

```
10 REM STOP
15 PRINT "(CLR)"
20 POKE 36878,15
25 FOR X=1 TO 15
30 FOR K=175 TO 240
35 S=S+1
40 POKE 36876,K
45 GET A$
50 IF A$<>"" THEN 65
55 NEXT K
60 S=S-166
65 POKE 36876,0
70 T=T+S
75 PRINT "SCORE"S,"TOTAL"T
80 S=0
85 FOR K=1 TO 500
90 NEXT K
95 NEXT X
100 END

READY.
```

GAME 29

PROGRAMMER

Description:

Programming in BASIC (or in any other language) requires that you pay attention and be careful. A misplaced comma or a misspelled word will result in a great big BUG!

The object of Programmer is to see how well you can copy BASIC statements. Six different statements are randomly presented and you must copy each statement presented exactly as shown. If you make a error, the statement will be presented again.

Here is how to interpret the time that is printed at the end of the program:

Time	Meaning (type of programmer)
0 - 25	A+
26 - 30	A
31 - 35	A−
36 - 40	B+
41 - 45	B
46 - 50	B−
51 - 55	C+
56 - 60	C
<60	Tenderfoot

Sample Run:

```
RUN
 IF N=1Ø THEN GOTO 5Ø
 ? IF N=1Ø THAN GOTO 5Ø
```

(An error is made.)

```
 IF N=1Ø THEN GOTO 5Ø
? IF N=1Ø THEN GOTO 5Ø
 S=S+1
?S=S+1
 FOR X=1 TO 1Ø
? FOR X= TO 1Ø
 GO TO 11Ø
? GOTO 11Ø
 INPUT N
? INPUT N
 PRINT ''SCORE=''S
? PRINT ''SCORE=''S
TIME= 37.8833333
```

(The statement is repeated.)

PROGRAM:

```
10 REM PROGRAMMER
15 FOR X=1 TO 6
20 READ W$(X)
25 NEXT X
30 DATA FOR X=1 TO 10
35 DATA GOTO 110
40 DATA INPUT N
45 DATA S=S+1
50 DATA PRINT "SCORE="S
55 DATA IF A$<>"" THEN 55
60 S=TI
65 FOR X=1 TO 6
70 R=INT(RND(1)*6+1)
75 IF V(R)=1 THEN 70
80 V(R)=1
85 PRINT
90 PRINT TAB(2);W$(R)
95 PRINT
100 INPUT S$
105 IF S$=W$(R) THEN 115
110 GOTO 85
115 NEXT X
120 F=TI
125 PRINT "TIME="(F-S)/60
130 END

READY.
```

GAME 30

ROCK, SCISSORS, PAPER

Description:

The 3 rules for this very old game are:

1. ROCK breaks SCISSORS
2. SCISSORS cut PAPER
3. PAPER covers ROCK

The game consists of 15 rounds. You play against the computer; ties are played over. For each play, enter ROCK, SCISSORS, or PAPER. Your score (0 to 15) is printed at the end of the game. Watch your spelling!

Sample Run:

```
RUN
ROCK SCISSORS PAPER
? PAPER
COMPUTER PICKS ROCK
YOU WIN!

ROCK SCISSORS PAPER
? ROCK
THE COMPUTER PICKS PAPER
OUCH!!
.
.
.
SCORE= 7
```

PROGRAM:

```
10 REM ROCK,SCISSORS,PAPER
15 READ A$(1),A$(2),A$(3)
20 DATA ROCK,SCISSORS,PAPER
25 FOR X=1 TO 15
30 PRINT
35 PRINT A$(1)" "A$(2)" "A$(3)
40 INPUT N$
45 C=INT(RND(1)*3+1)
50 PRINT "COMPUTER PICKS "A$(C)
55 IF A$(C)<>N$ THEN 70
60 PRINT "TIE"
65 GOTO 30
70 IF C=1 AND N$=A$(3) THEN 95
75 IF C=2 AND N$=A$(1) THEN 95
80 IF C=3 AND N$=A$(2) THEN 95
85 PRINT "OUCH!!"
90 GOTO 105
95 PRINT "YOU WIN!"
100 S=S+1
105 NEXT X
110 PRINT "SCORE="S
115 END

READY.
```

GAME 31

AROUND THE CLOCK

Description:

Around the Clock is a simulated dice game using a single die. Press the RETURN key to "toss" the die. Each toss will lower your score by one point.

For the first round, you must toss a 1; for the second round, you must toss a 2. This procedure is repeated until you have tossed a 1, 2, 3, 4, 5, and 6 in that order.

What your score means:

85 to 94 or 0 to 30:	Legion of Merit
80 to 74 or 31 to 35:	Gold Medal
75 to 79 or 36 to 40:	Silver Medal
70 to 74 or 41 to 45:	Bronze Medal
65 to 69 or 46 to 50:	a nice little ribbon
60 to 64 or 51 to 55:	pat on the back
56 to 59:	disqualified

Sample Run:

```
RUN
TOSS 1
?                                   (press RETURN key)
YOU TOSSED 5

TOSS 1
?
YOU TOSSED 4
```

```
TOSS 1
?
YOU TOSSED 1

TOSS 2
.
.
.
.
.
TOSS 6
?
YOU TOSSED 6

FINISHED!
SCORE= 58
```

PROGRAM:

```
10 REM AROUND THE CLOCK
15 X=1
20 PRINT
25 PRINT "TOSS"X
30 INPUT T$
35 R=INT(RND(1)*6+1)
40 PRINT "YOU TOSSED"R
45 S=S+1
50 IF R=X THEN 60
55 GOTO 20
60 X=X+1
65 PRINT
70 IF X=7 THEN 80
75 GOTO 20
80 PRINT "FINISHED!"
85 PRINT "SCORE="100-S
90 END

READY.
```

GAME 32

MAZE

Description:

In the Maze game, the computer creates your own personal maze each time you run the program.

Imagine you are in a long hall or alley. At the end of the alley you can go either right or left. If you go in the correct direction, you move on to the next alley and thus continue through the maze. If you move in the wrong direction, you are told that you have made a mistake.

The maze game is over when you are able to go through the maze without making a mistake. A perfect score is 0 and means that you are an awfully lucky player at a sport designed primarily for mice.

Sample Run:

```
RUN

ALLEY 1                 (Enter L if you want to go Left, and
L OR R? L               R if you want to go Right.)
INCORRECT

ALLEY 1                 (The next alley will be presented only
L OR R? R               when you have entered the correct
CORRECT!                direction.)

ALLEY 2
L OR R? R
CORRECT!
```

```
.
.
.
.
.
ALLEY 6
L OR R? L
CORRECT!

ERRORS= 9
```

PROGRAM:

```
10 REM MAZE
15 FOR X=1 TO 6
20 IF INT(RND(1)*2)=1 THEN 35
25 M$(X)="L"
30 GOTO 40
35 M$(X)="R"
40 NEXT X
45 A=1
50 PRINT "(CLR)(DN)(DN)(DN)(DN)(DN)(DN)"
55 PRINT "ALLEY"A
60 INPUT "L OR R";D$
65 IF D$=M$(A) THEN 85
70 E=E+1
75 PRINT "INCORRECT"
80 GOTO 95
85 PRINT "CORRECT!"
90 A=A+1
95 FOR K=1 TO 750
100 NEXT K
105 IF A=7 THEN 115
110 GOTO 50
115 IF E=0 THEN 135
120 T=T+E
125 E=0
130 GOTO 45
135 PRINT
140 PRINT "ERRORS="T
145 END

READY.
```

GAME 33

SHOOTING GALLERY

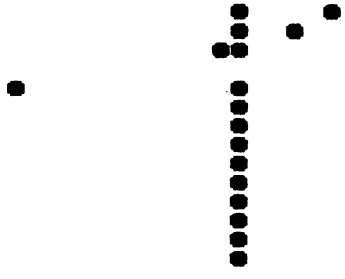
Description:

Step right up to the Shooting Gallery and see how many targets you can hit. There are 20 targets and you are given 20 shots. Press the RETURN key to shoot. If you hit the target, it will disappear from the screen. If you miss, a red capsule-shaped dot ahead of or behind the target will indicate whether your shot was too early or too late. The total number of hits (0 to 20) is printed at the end of the game.

As with many aracade-type games, Shooting Gallery uses pokes to create the graphics. As an example, the first target is made by POKE 7716,81. The 81 indicates that a dot is to be printed at screen location 7716. As it is, screen location 7716 is the 15th column of the second row. The POKE 38436,0 colors the target (capsule-shaped dot) at this location black. Consult your VIC 20 manual (pages 139 to 144) to learn more about screen display codes.

Sample Run:

RUN



(A miss is indicated by a red dot.)

(A black target disappears from the screen when hit.)

PROGRAM:

```
10 REM SHOOTING GALLERY
15 PRINT "(CLR)"
20 FOR X=0 TO 418 STEP 22
25 POKE 7716+X,81
30 POKE 38436+X,0
35 NEXT X
40 FOR X=0 TO 418 STEP 22
45 FOR K=0 TO 20
50 POKE 7702+X+K,81
55 POKE 38422+X+K,0
60 GET A$
65 IF A$<>"" THEN 85
70 IF K=14 THEN 80
75 POKE 38422+X+K,1
80 NEXT K
85 IF K<>14 THEN 105
90 POKE 38422+X+K,1
95 S=S+1
100 GOTO 110
105 POKE 38422+X+K,2
110 FOR L=1 TO 1500
115 NEXT L
120 NEXT X
125 PRINT "HITS="S
130 END

READY.
```

GAME 34

MAGIC SQUARE

Description:

The goal of this game is to enter the numbers 1 through 9 so that each row sum and each column sum equals 15 (and the total sum equals 45). A more difficult task is to enter a square so that each row sum, each column sum, and each diagonal sum equals 15.

You enter the numbers row by row, three numbers being entered for each row. For each square, the numbers 1 through 9 should only be used once.

Sample Run:

```
RUN
SQUARE 1
1    2    3    6
4    5    6    15
7    8    9    24

12   15   18   45

ROW 1? 4,2,9
ROW 2? 8,6,1
ROW 3? 7,3,5

SQUARE 2
4    2    9    15
8    6    1    15
7    3    5    15
```

```
19  11  15  45

ROW 1?
```

PROGRAM:

```
10 REM MAGIC SQUARE
15 READ A,B,C,D,E,F,G,H,I
20 DATA 1,2,3,4,5,6,7,8,9
25 PRINT "(CLR)"
30 PRINT
35 PRINT "SQUARE"N+1
40 PRINT A;TAB(4);B;TAB(8);C;TAB(12);A+B+C
45 PRINT D;TAB(4);E;TAB(8);F;TAB(12);D+E+F
50 PRINT G;TAB(4);H;TAB(8);I;TAB(12);G+H+I
55 PRINT
60 T=A+B+C+D+E+F+G+H+I
65 PRINT A+D+G;TAB(4);B++E+H;TAB(8);C+F+I;TAB(12);T
70 PRINT
75 INPUT "ROW 1";A,B,C
80 INPUT "ROW 2";D,E,F
85 INPUT "ROW 3";G,H,I
90 N=N+1
95 GOTO 25
100 END

READY.
```

GAME 35

QUARTERBACK

Description:

This is your chance to think like a quarterback. The object of Quarterback is to score a TOUCHDOWN. You are given four plays to make a first down. This is achieved by moving the ball 10 yards. Each time you make a first down, you are given four additional downs. The game is over when you fail to make a first down or when you score a TOUCHDOWN. If you fail to score a first down, your score is the yard line at which you lost control of the ball.

You run a play by entering P (for pass) or R (for run). If you pass the ball, you can gain between −10 to 25 yards. If you run the ball, you can gain between −2 to 12 yards. What is the best playing strategy? Play Quarterback and find out.

Sample Run:

```
RUN
R OR P? R
GAIN Ø
YARDS FOR FIRST 1Ø

R OR P? P
GAIN-5
R OR P? P          (A first down was made and the
YARD LINE 38       ball is now on the 38 yard line.)

R OR P? R
YARD LINE 49
.
```

```
.
.
.
.
R OR P? P
GAIN 3
YARDS FOR FIRST 7

R OR P? R
GAIN 4
YARDS FOR FIRST 3

R OR P? R
GAIN Ø
YARDS FOR FIRST 3

R OR P? R
GAIN 2
YARDS FOR FIRST 1
LOST BALL AT 96
```

PROGRAM:

```
10 REM QUARTERBACK
15 B=20
20 T=0
25 FOR X=1 TO 4
30 R=INT(RND(1)*2)
35 PRINT
40 INPUT "R OR P";D$
45 G=T
50 IF D$="R" AND R=0 THEN T=T+INT(RND(1)*7-2)
55 IF D$="R" AND R=1 THEN T=T+INT(RND(1)*12)
60 IF D$="P" AND R=0 THEN T=T+INT(RND(1)*25)
65 IF D$="P" AND R=1 THEN T=T+INT(RND(1)*10-10)
70 IF T+B>99 THEN 115
75 IF T>9 THEN 100
80 PRINT "GAIN"T-G
85 PRINT "YARDS FOR FIRST"10-T
90 NEXT X
95 GOTO 125
100 B=B+T
105 PRINT "YARD LINE"B
110 GOTO 20
115 PRINT "TOUCHDOWN!"
120 GOTO 130
125 PRINT "LOST BALL AT"B+T
130 END

READY.
```

GAME 36

MATH WHIZ

Description:

To become a MATH WHIZ (the absolute high ranking in this game) you must answer 15 easy math problems. The following is an example of the type of math problem that is given:

```
6 + 1 - Ø + 4
```

Simple? Well, not exactly. For the first problem you have approximately 10 seconds to enter the correct answer. For each of the following problems, the time to enter the correct answer becomes less and less. This is a tricky little game, so pay attention and watch those pluses and minuses.

Sample Run:

```
RUN
5 + 3 - 2 + 3 (Answer was not entered in time.)
TOO LATE
LEVEL Ø

RUN
7 + 1 - Ø + 1 (Incorrect answer was entered and
ERROR           the game ends.)
LEVEL Ø

RUN
6 + 2 - Ø + 3  (The correct answer was entered,
                so the computer presents
Ø + 4 - 3 + 6  another problem.)
```

```
2 + 3 - Ø - Ø

5 + 2 - Ø + 1

3 + 1 - Ø + 6

2 + 5 - 2 + 6

1 + 2 - Ø + 7

3 + 7 - 4 + Ø

4 + 6 - 3 + 2

2 + 2 - Ø + 1

3 + 5 - 4 + 2
ERROR
LEVEL 1Ø
```

(Incorrect answer is entered.)

PROGRAM:

```
10 REM MATH WHIZ
15 M=500
20 FOR X=1 TO 15
25 A=INT(RND(1)*8)
30 B=INT(RND(1)*7+1)
35 C=INT(RND(1)*8)
40 D=INT(RND(1)*8)
45 PRINT "(CLR)(DN)(DN)(DN)(DN)(DN)(DN)(DN)"
50 PRINT A"+"B"-"C"+"D
55 FOR K=1 TO M
60 IF A$>"" THEN 75
65 GET A$
70 IF A+B-C+D<10 AND A$>"" THEN 105
75 IF A$="" THEN 90
80 GET B$
85 IF A$>"" AND B$>"" THEN 105
90 NEXT K
95 PRINT "TOO LATE"
100 GOTO 145
105 IF A+B-C+D<>VAL(A$+B$) THEN 140
110 M=M-30
115 A$=""
120 B$=""
125 NEXT X
130 PRINT "MATH WHIZ"
135 GOTO 150
140 PRINT "ERROR"
145 PRINT "LEVEL"X-1
150 END

READY.
```

GAME 37

DECODING

Description:

This game uses ASCII codes (American Standard Code for Information Interchange) to determine your ability to decipher a series of numbers into their letter equivalents. Each letter of the alphabet has an ASCII code equivalent. The letter A is represented by the number 65. Using the CHR$ function, CHR$(65) is equivalent to the letter A. The relationship between letters and ASCII numbers follows this format:

```
A=65  B=66  C=67  D=68  E=69  .  .  .
X=88  Y=89  Z=90
```

In this game 30 ASCII numbers are given, and you are required to give the letter equivalent for each ASCII number presented. If you give an incorrect answer, the computer will tell you whether your response was TOO low (65 was presented and you answered Z) or TOO HIGH (a 90 was presented and you answered A).

You are only given 30 tries in this game, and every incorrect answer counts as a try. Each letter is represented until it is answered correctly. A perfect score is 30.

Sample Run:

```
RUN
CODE: 75
LETTER? L
TOO HIGH

CODE: 75
```

```
LETTER? K
CORRECT!

CODE: 66
LETTER? B
CORRECT

CODE: 71
LETTER? E
TOO LOW

CODE: 71
LETTER? G
CORRECT!
.
.
.
.
.
CODE: 85
LETTER? U
CORRECT!
SCORE= 17
```

PROGRAM:

```
10 REM DECODING
15 FOR X=1 TO 30
20 R=INT(RND(1)*26+65)
25 L$=CHR$(R)
30 PRINT
35 PRINT "CODE:"R
40 INPUT "LETTER";A$
45 IF A$=L$ THEN 70
50 IF A$<L$ THEN PRINT   "TOO LOW"
55 IF A$>L$ THEN PRINT "TOO HIGH"
60 X=X+1
65 GOTO 30
70 PRINT "CORRECT!"
75 S=S+1
80 NEXT X
85 PRINT
90 PRINT "SCORE="S
95 END

READY.
```

GAME 38

RACE CAR

Description:

You can follow the speed of the race car by watching the graphic character in the center of the screen. Your task is to accelerate as fast as possible and complete the run.

As soon as you see the first graphic character appear, you can accelerate by pressing the letters of the alphabet (this is a smart car) in the proper sequence beginning with the letter A. The more letters you press, the faster the car will go. However, if you do not press the letters in the proper sequence, you will receive the message FALSE START and the race is over.

Your time to complete the run is printed at the end of the race. Here's how to interpret your time:

Time	Meaning
<10	Grand Prix Driver
10 - 12	Indy 500 Driver
13 - 15	Stock Car Driver
16 - 20	Go-cart Driver
>20	Get a Bike

Sample Run:

```
RUN
▮
▮
```

(The graphic character moves faster as you press the letters of the alphabet in the correct sequence.)

```
TIME 15.816667
```

PROGRAM:

```
10 REM RACE CAR
15 M=130
20 S=TI
25 FOR X=1 TO 80
30 PRINT
35 PRINT TAB(11);CHR$(161)
40 FOR K=1 TO M
45 GET A$
50 IF A$<>"" THEN 65
55 NEXT K
60 GOTO 75
65 M=M-5
70 IF CHR$(65+N)<>A$ THEN 100
75 IF CHR$(65+N)=A$ THEN N=N+1
80 NEXT X
85 F=TI
90 PRINT "TIME"(F-S)/60
95 GOTO 105
100 PRINT "FALSE START!"
105 END

READY.
```

GAME 39

SORT

Description:

The game of Sort is a test of your dictionary skills. Indeed, this might be one of the few, if not the only, contest in which you attempt to break the dictionary Sort record.

The computer begins by printing five strings of letters, each string having ten letters. The letters in each string are G, H, I, J, and K. The task is to arrange the strings in the correct alphabetical order.

When the computer prints ENTER STRING 1, you must enter (exactly) the string that is printed on the screen that has the lowest alphabetical ranking. For example, the string GJJGHGIKKJ would be entered before the string IKKIJKHGKJ.

If you enter a string out of order, or enter a string that is not exactly the same as a string shown on the screen, the computer will indicate what string should have been entered and the game will be over.

A time of less than 45 seconds is a world record; less than 55 seconds is fantastic; less than 65 seconds is very good; less than 75 seconds is good; and a time of less than 90 seconds means that you must hit the old Sort practice field.

Sample Run:

```
RUN
1  KKIIIHHHHH
2  JJHGKKGJGJ
3  IGGHIKIHKK
4  JJGIGJHJGH
5  GJGGGHHHJI
```

```
STRING 1
? GJGGGHHHJI

STRING 2
? IGGHIKIHKK

ENTER STRING 3
? JJGIGJHJGH

ENTER STRING 4
? JJHGKKGJGJ

ENTER STRING 5
? KKIIIHHHHH
TIME= 56.5833333
```

PROGRAM:

```
10 REM SORT
15 FOR X=1 TO 5
20 FOR Y=1 TO 10
25 L$(X)=L$(X)+CHR$(INT(RND(1)*5+71))
30 NEXT Y
35 PRINT X;L$(X)
40 NEXT X
45 S=TI
50 FOR X=1 TO 4
55 FOR Y=1 TO 4
60 IF L$(Y)<L$(Y+1) THEN 80
65 H$=L$(Y)
70 L$(Y)=L$(Y+1)
75 L$(Y+1)=H$
80 NEXT Y
85 NEXT X
90 FOR X=1 TO 5
95 PRINT
100 PRINT "ENTER STRING"X
105 INPUT A$
110 IF A$<>L$(X) THEN 135
115 NEXT X
120 F=TI
125 PRINT "TIME="(F-S)/60
130 GOTO 140
135 PRINT "ANSWER: "L$(X)
140 END

READY.
```

GAME 40

FINGERQUICK

Description:

There are persons with fast legs, and there are fast writers, fast readers, and fast talkers, but Fingerquick is a true test of speed (if you are primarily interested in microcomputers).

After you run Fingerquick, there will be a pause and the computer will print a number between 0 and 9. Once the number is printed, your task is to press the key indicated as many times as possible within (approximately) a 5-second period. If you play more than once, be sure the screen is clear of numbers before typing the RUN command.

A score of 40 means that you are a world-class Fingerquick player; a score above 30 is extremely good; a score of 20 or more is good; and a score of less than 20 means that you are definitely in need of more Fingerquick playing practice.

Sample Run:

```
RUN

PRESS: 3    (The key is pressed as many times as
             possible within a 5-second period.)
SCORE= 27
```

PROGRAM:

```
10 REM FINGERQUICK
15 PRINT "(CLR)(DN)(DN)(DN)(DN)(DN)(DN)(DN)"
20 R$=CHR$(INT(RND(1)*10+48))
25 FOR X=1 TO 500
30 NEXT X
35 PRINT "PRESS: "R$
40 FOR X=1 TO 750
45 GET A$
50 IF A$=R$ THEN S=S+1
55 A$=""
60 NEXT X
65 PRINT
70 PRINT "SCORE="S
75 END

READY.
```

GAME 41

COMPUTER BRAIN

Description:

Have you ever wondered how a computer thinks? If you have (and even if you haven't), now is your chance to play Computer Brain.

The computer will pick a number between 1 and 1000, and then count up to that number. The computer will begin counting after the word GO is printed, and stop counting when the word STOP is printed.

Based on the delay between the words GO and STOP, you must determine the number picked by the computer. If your score has a minus sign, this means that your estimate of the number picked by the computer was too low; and a positive score means that your estimate was too high. Here's how to interpret your score:

Score	Meaning
50 to −50	Computer Brain
100 to −100	Calculator Brain
200 to −200	Abacus Brain
300 to −300	You have a Brain

Sample Run:

```
RUN
 GO
 STOP
? 900
SCORE=-32
```

(The number picked by the computer was 932.)

PROGRAM:

```
10 REM COMPUTER BRAIN
15 PRINT "(CLR)(DN)(DN)(DN)(DN)(DN)(DN)(DN)"
20 R=INT(RND(1)*1000+1)
25 FOR X=1 TO 1000
30 NEXT X
35 PRINT TAB(5)"GO"
40 FOR K=1 TO R
45 NEXT K
50 PRINT TAB(5)"STOP"
55 INPUT A
60 PRINT "SCORE="A-R
65 END

READY.
```

GAME 42

INTERCEPT

Description:

No collection of games is complete without at least one missile-type game. The game of Intercept consists of shooting down as many enemy targets as possible within a 1-minute period.

The enemy targets appear at the top of the screen, and the missile to be fired each round appears at the bottom of the screen. Before firing, you must line up the missile directly below the target or the missile will not hit the target when fired. You move the missile by pressing the SPACE key. You fire the missile by pressing the key with the upward-pointing arrow.

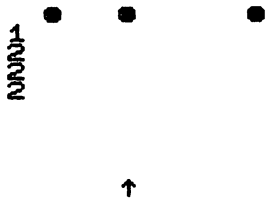
Two important rules to remember: First, you can only move the missile to the right. If you move the missile beyond the enemy target, you cannot move back to hit the target. Second, if you miss a target, the target will remain on the screen. However, even though there are other targets on the screen, you will only be given credit for hitting the last target presented on the screen.

Here's how to interpret your score:

Score	Meaning
13 to 20	Ace
9 to 12	First Class
5 to 8	Cadet
<5	No Rank

Sample Run:

```
RUN
```



(An enemy target will appear at the top of the screen.)

(The missile is moved by pressing the SPACE key.)

(When the missile is lined up underneath the target, press the upward-pointing arrow key to fire.)

PROGRAM:

```
10 REM INTERCEPT
15 TF=TI
20 PRINT "(CLR)"
25 BA=8142
30 R=INT(RND(1)*17+5)
35 POKE 7680+R,81
40 POKE 38400+R,0
45 POKE BA,30
50 POKE BA+30720,0
55 A$=""
60 GET A$
65 IF A$=CHR$(32) THEN BA=BA+1
70 POKE BA-1,30
75 POKE BA+30719,1
80 IF A$="^" THEN 95
85 IF TI-TF>3600 THEN 145
90 GOTO 45
95 FOR X=0 TO 462 STEP 22
100 POKE BA-X,30
105 POKE BA+30720-X,0
110 FOR L=1 TO 25
115 NEXT L
120 POKE BA+30720-X,1
125 NEXT X
130 IF 7680+R=BA-X+22 THEN S=S+1
135 PRINT S
140 GOTO 25
145 END

READY.
```

GAME 43

CONCENTRATION

Description:

The letters A, B, C, D, E, and F are randomly assigned to 50 different storage locations, beginning with L$(10) and ending with L$(59). For each round, the numbers of the 50 storage locations are printed. You then try to make a match by entering two of the storage numbers. After each storage number is entered, the letter in that storage area is shown. One point is given for every correct match. Once a match is made, the number 99 replaces the numbers of the storage areas used in the match.

You are given 20 chances to match. You cannot match by selecting numbers less than 10 or greater than 59; nor by selecting the same storage area for a given match.

The game truly requires concentration (and a little bit of luck)!

Sample Run:

```
RUN
 10  11  12  13  14
 15  16  17  18  19
 20  21  22  23  24
 25  26  27  28  29
 30  31  32  33  34
 35  36  37  38  39
 40  41  42  43  44
 45  46  47  48  49
 50  51  52  53  54
```

```
 55 56 57 58 59
FIRST CHOICE? 10
      10 =B
SECOND CHOICE? 13
      13 =B
     MATCH!
 99 11 12 99 14
 15 16 17 18 19
 20 21 22 23 24
 25 26 27 28 29
 30 31 32 33 34
 35 36 37 38 39
 40 41 42 43 44
 45 46 47 48 49
 50 51 52 53 54
 55 56 57 58 59
FIRST CHOICE? 12
      12 =E
SECOND CHOICE?
      14 =B
     NO MATCH
```

(The number 99 replaces the storage areas correctly matched.)

PROGRAM:

```
10 REM CONCENTRATION
15 DIM N(59),L$(59)
20 FOR X=10 TO 59
25 N(X)=X
30 L$(X)=CHR$(INT(RND(1)*6+65))
35 NEXT X
40 FOR X=1 TO 20
45 PRINT "(CLR)"
50 FOR K=10 TO 59
55 PRINT TAB(2);N(K);
60 NEXT K
65 INPUT "FIRST CHOICE";FC
70 PRINT TAB(5);FC"="L$(FC)
75 INPUT "SECOND CHOICE";SC
80 PRINT TAB(5);SC"="L$(SC)
85 IF L$(FC)<>L$(SC) THEN 120
90 IF FC<10 OR FC=SC THEN 120
95 S=S+1
100 N(FC)=99
105 N(SC)=99
110 PRINT TAB(5)"MATCH!"
115 GOTO 125
120 PRINT TAB(5)"NO MATCH"
125 FOR L=1 TO 2500
130 NEXT L
135 FC=0
140 NEXT X
145 PRINT "(DN)(DN)SCORE="S
150 END

READY.
```

GAME 44

PICKLE

Description:

The game of Pickle begins by your being given a number between 0 and 9. If this first number is 0 or 9, you lose and the game is over. Pickle can be a extremely short game. However, if this number is 2 or 7, then you win either 2 points or 7 points.

What happens if the first number is 1, 3, 4, 5, 6, or 8? Good question. If this happens, you must keep pressing the return key until you match the number drawn (and win that number of points) or until you draw a 7 (and the game is over).

Although the history of the game of Pickle is vague, it is customary to say "pickle" after losing.

The highest recorded score by a certified Pickle player was 57. Note, however, that this was accomplished after losing 1,134 games of Pickle. The Pickle player who achieved this score is now retired.

Sample Run:

```
RUN
PRESS KEY?                    (press RETURN key)
NUMBER 4
PRESS FOR MATCH?              (press RETURN key)

DRAW 8                        (the number was
PRESS FOR MATCH?              not matched.)
DRAW 9
PRESS FOR MATCH?
DRAW 4
```

```
YOU WIN!

PRESS KEY?
NUMBER 7
YOU WIN?

PRESS KEY?
NUMBER 8
PRESS FOR MATCH?
DRAW 1
PRESS FOR MATCH?
DRAW 6
PRESS FOR MATCH?
DRAW 7
YOU LOSE

SCORE= 11
```

PROGRAM:

```
10 REM PICKLE
15 PRINT
20 INPUT "PRESS KEY";P$
25 N=INT(RND(1)*10)
30 PRINT "NUMBER"N
35 IF N=0 OR N=9 THEN 90
40 IF N=2 OR N=7 THEN 75
45 INPUT "PRESS FOR MATCH";R$
50 M=INT(RND(1)*10)
55 PRINT "DRAW"M
60 IF N=M THEN 75
65 IF M=7 THEN 90
70 GOTO 45
75 PRINT "YOU WIN!"
80 S=S+N
85 GOTO 15
90 PRINT "YOU LOSE"
95 PRINT
100 PRINT "SCORE="S
105 END

READY.
```

GAME 45

LINE CHECKERS

Description:

There are nine positions in line checkers. The object is to move the four + characters to the bottom four positions (6,7,8, and 9) in the fewest possible moves.

The rules are:

1. You must move a + or * to an open position. You cannot move a character to a position that is already occupied.
2. You can jump to an open position, but you can only jump over one or two characters.

Sample Run:

```
RUN
 1 +
 2 +
 3 +
 4 +
 5
 6 *
 7 *
 8 *
 9 *
MOVES TAKEN Ø
WHERE TO WHERE? 4,5
 1 +
 2 +
 3 +
```

(A + is moved from position 4 to position 5.)

```
 4
 5 +
 6 *
 7 *
 8 *
 9 *
MOVES TAKEN 1
WHERE TO WHERE? 7,4
 1 +
 2 +
 3 +
 4 *
 5 +
 6 *
 7
 8 *
 9 *
```

PROGRAM:

```
10 REM LINE CHECKERS
15 FOR X=1 TO 9
20 READ T$(X)
25 NEXT X
30 DATA +,+,+,+,"",*,*,*,*
35 FOR X=1 TO 9
40 PRINT
45 PRINT TAB(5);X;T$(X)
50 NEXT X
55 PRINT "MOVES TAKEN"M
60 IF T$(6)+T$(7)+T$(8)+T$(9)="++++" THEN 105
65 INPUT "WHERE TO WHERE";A,Z
70 IF T$(A)="" OR T$(Z)<>"" THEN 90
75 IF ABS(A-Z)>3 THEN 90
80 T$(Z)=T$(A)
85 T$(A)=""
90 M=M+1
95 PRINT
100 GOTO 35
105 PRINT "FINISHED!!"
110 END

READY.
```

GAME 46

BIRTHDAY

Description:

On what day of the week were you born? The game of Birthday will help you determine if you were born on a Monday, Tuesday, etc.

To play Birthday you must give the computer some preliminary information. You must first enter a code for the month in which you were born. The codes are:

1 March	7 September
2 April	8 October
3 May	9 November
4 June	10 December
5 July	11 January
6 August	12 February

After you have entered the month code, enter the day of the month (e.g., 1,2, etc.) on which you were born. Finally, enter the year in which you were born.

You have seven guesses to name the day of the week on which you were born. Misspellings will count as guesses so be careful. If you fail to guess the name of your birthdate in seven attempts, the computer will inform you that you have NO BIRTHDAY. If this happens, you better play again.

Sample Run:

```
RUN
DATA:
MONTH CODE? 4
MONTH DATE? 12
```

```
YEAR? 1943
DAY BORN? WEDNESDAY
DAY BORN? SUNDAY
DAY BORN? MONDAY
DAY BORN? TUESDAY
DAY BORN? SATURDAY
CORRECT!!
GUESSES= 5
```

PROGRAM:

```
10 REM BIRTHDAY
15 FOR X=0 TO 6
20 READ D$(X)
25 NEXT X
30 DATA SUNDAY,MONDAY,TUESDAY,WEDNESDAY,THURSDAY,FRIDAY
,SATURDAY
35 PRINT "(CLR)"
40 PRINT "DATA:"
45 INPUT "MONTH CODE";T1
50 INPUT "MONTH DATE";T2
55 INPUT "YEAR";Y
60 T3=INT(Y/100)
65 T4=Y-(T3*100)
70 S=2.6*T1-.2+T2+T4+INT(T3/4)+INT(T4/4)-2*T3
75 L=INT(S-7*INT(S/7))
80 FOR X=1 TO 7
85 INPUT "DAY BORN";B$
90 IF B$=D$(L) THEN 105
95 NEXT X
100 GOTO 125
105 PRINT "CORRECT!!"
110 PRINT "GUESSES="X
115 GOTO 130
125 PRINT "NO BIRTHDAY"
130 END

READY.
```

GAME 47

THE RUN

Description:

The Run is a game of directions. To complete the trek across the screen you must follow a series of simple directions. Whenever the computer prints the symbol < (pointing left), you must press the L key. Whenever the computer prints the = symbol (indicating center), you must press the C key. Finally, whenever the computer prints the symbol > (pointing right), you must press the R key.

As with many running events, The Run gets more difficult as you advance across the screen. This occurs because once a symbol is flashed, you have a limited amount of time to press the appropriate key . . . and this time shortens as you continue your run.

Here's how to interpret your score:

Score	Meaning
18 - 20	World Class
15 - 17	Olympic Hopeful
12 - 14	State Champion
9 - 11	City Champion
6 - 8	Local Champion
3 - 5	Still in Training
0 - 2	Injury List

Sample run:

```
RUN
```

● < (The symbol is pointing left, so the L key is pressed.)

● > (The symbol is pointing right, so the R key is pressed.)

● > (The symbol is pointing right, so the R key is pressed.)

● = (The = symbol appears so, the C key is pressed.)

● < (The game is over because the R key was pressed instead of the L key.)

PROGRAM:

```
10 REM THE RUN
15 T=1145
20 FOR X=1 TO 20
25 PRINT "(CLR)"
30 POKE 7900+K,81
35 POKE 38620+K,0
40 D=INT(RND(1)*3+60)
45 POKE 7902+K,D
50 POKE 38622+K,0
55 FOR L=1 TO T
60 GET A$
65 IF A$<>"" THEN 80
70 NEXT L
75 GOTO 115
80 IF D=60 AND A$="L" THEN 100
85 IF D=61 AND A$="C" THEN 100
90 IF D=62 AND A$="R" THEN 100
95 GOTO 115
100 T=T-55
105 K=K+1
110 NEXT X
115 PRINT
120 PRINT "SCORE="X-1
125 END

READY.
```

GAME 48

GOBBLER

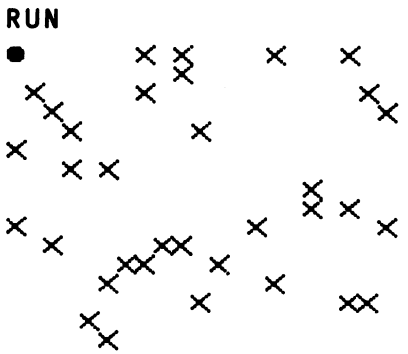
Description:

The object of this game is to move the Gobbler left, right, up, or down in order to devour as many X's as possible. You move the gobbler as follows:

1. press the A key to move left;
2. press the S key to move right;
3. press the K key to move up;
4. press the L key to move down.

Each time the gobbler gobbles an X, you receive one point. There is a time limit of approximately 30 seconds so you should attempt to move where the X's are and gobble as quickly as possible.

Sample Run:



(Move the gobbler to eat as many X's as possible.)

PROGRAM:

```
10 REM GOBBLER
15 PRINT "(CLR)"
20 FOR X=0 TO 505
25 IF INT(RND(1)*50)<46 THEN 35
30 POKE 7680+X,86
35 POKE 38400+X,0
40 NEXT X
45 K=7680
50 A=TI
55 GET A$
60 POKE K,81
65 POKE 30720+K,0
70 PK=K
75 IF A$="S" THEN K=K+1
80 IF A$="A" THEN K=K-1
85 IF A$="K" THEN K=K-22
90 IF A$="L" THEN K=K+22
95 IF PK<>K THEN POKE 30720+PK,1
100 IF K<7680 THEN K=7680
105 IF K>8185 THEN K=8185
110 IF PEEK(K)<>86 THEN 125
115 S=S+1
120 POKE 30720+K,1
125 IF TI-A>1800 THEN 135
130 GOTO 55
135 PRINT "(CLR)"
140 PRINT "SCORE="S
145 END

READY.
```

GAME 49

FLASH

Description:

Each round of Flash consists of four numbers being flashed for a fraction of a second on the screen. Once the four numbers disappear, you must input the numbers that appeared in the exact order in which they were shown.

If you correctly entered the numbers shown, there will be a brief pause and four new numbers will appear. However, these numbers will appear on the screen for a shorter period than the last four numbers shown. By the time you reach the end of Flash, the four numbers that appear will truly be little more than a flash on the screen.

Here's how to interpret your score:

Score	Meaning
2	The Flash
4 - 8	Quick Eyes
10 - 20	Attentive
22 - 30	OK
32 - 40	You are tired
42 - 50	You are asleep

Sample Run:

```
RUN
   9342          (Number will appear for a brief moment.)
? 9342           (The number is entered exactly as
   6579          it was shown.)
? 6579
```

```
   8410
 ? 8410
   .
   .
   .
   9308
 ? 9388
THE NUMBER WAS 9388
SPEED= 6
```

PROGRAM:

```
10 REM FLASH
15 N=500
20 FOR X=1 TO 4
25 D$=D$+CHR$(INT(RND(1)*10+48))
30 NEXT X
35 PRINT "(CLR)"
40 FOR L=1 TO 1500
45 NEXT L
50 PRINT "(CLR)(DN)(DN)(DN)(DN)(DN)(DN)(DN)(DN)(RT)(RT)
(RT)(RT)(RT)(RT)(RT)"D$
55 FOR K=1 TO N
60 NEXT K
65 PRINT "(CLR)(DN)(DN)(DN)(DN)(DN)(DN)(DN)(DN)(RT)(RT)
(RT)(RT)(RT)";
70 INPUT A$
75 IF A$<>D$ THEN 100
80 N=N-20
85 D$=""
90 IF N=20 THEN 105
95 GOTO 20
100 PRINT "THE NUMBER WAS ";D$
105 PRINT "SPEED="N/10
110 END

READY.
```

GAME 50

TREE CLIMB

Description:

You never would have thought that you would be competing against a computer in a tree-climbing contest, would you?

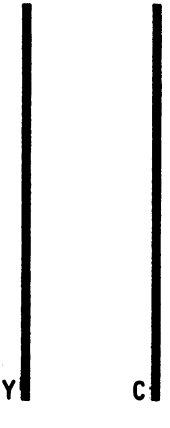
The tree-climbing race begins after the computer prints two vertical lines from the top to the bottom of the screen. These, I admit, are not much in the line of trees, but don't expect a work of art in a program less than 30 statements long.

You are represented at the bottom of the tree on the left by a Y, and the computer is represented by a C. As soon as the Y and C appear, the race begins. The first (Y or C) to reach the top is the winner. The winner's time is printed at the end of the race.

The world record for the Tree Climb (based on the few living tree-climbing contestants) is the amazing time of 17.6833333. Don't expect a time anywhere near this, since this effort was the result of many tree-climbing practice sessions.

Sample Run:

```
RUN
```



PROGRAM:

```
10 REM TREE CLIMB
15 PRINT "(CLR)"
20 FOR X=0 TO 22
25 POKE 7687+X*22,97
30 POKE 38407+X*22,0
35 POKE 7694+X*22,97
40 POKE 38414+X*22,0
45 NEXT X
50 Y=8170
55 C=8177
60 F=TI
65 POKE Y,25
70 POKE 30720+Y,0
75 POKE C,3
80 POKE 30720+C,0
85 R=INT(RND(1)*2)
90 IF R=0 THEN Y=Y-22
95 IF R=1 THEN C=C-22
100 IF Y=7686 THEN 135
105 IF C=7693 THEN 145
110 FOR K=1 TO INT(RND(1)*600+10)
115 NEXT K
120 IF R=0 THEN POKE 30742+Y,1
125 IF R=1 THEN POKE 30742+C,1
130 GOTO 65
135 PRINT "YOUR TIME="(TI-F)/60
140 GOTO 150
145 PRINT "COMPUTER'S TIME="(TI-F)/60
150 END

READY.
```

HATE TO KEY-IN PROGRAM LISTINGS?

Spend $7.95 for a ready-to-run tape of the programs in *VIC 20: 50 Easy-to-Run Computer Games* and you'll avoid spending hours tracking down a bug accidentally typed into a stalled or crashed program (it can happen to *anybody*)!

Ask for No. 22287 .....$7.95

Special combination package — book and tape together!

Ask for No. 22288 .....$17.95

Use the handy order form and send to the address below or call the Sams Order Desk at 800-428-3696 or 317-298-5566 and charge it to your MasterCard or Visa account.

Ask about other Sams book/tape combinations for the VIC 20, Commodore 64, and other microcomputers too, or contact your local Sams dealer.

Howard W. Sams & Co., Inc.
4300 W. 62nd Street — P.O. Box 7092
Indianapolis, IN 46206

Catalog No.	Qty.	Price	Total	Catalog No.	Qty.	Price	Total

	Subtotal
☐ Check ☐ Money Order ☐ Visa	Add local tax where applicable
☐ MasterCard	Add Handling Charge 2.00
	Total Amount Enclosed

Account Number _____ Expires _____

Name (print) _____

Signature _____

Address _____

City _____ State _____ Zip _____

Prices subject to change without notice. Offer good in USA only. In Canada, contact Lenbrook Electronics, Markham, Ontario L3R 1H2.

X0462

TO THE READER

Sams Computer books cover Fundamentals — Programming — Interfacing — Technology written to meet the needs of computer engineers, professionals, scientists, technicians, students, educators, business owners, personal computerists and home hobbyists.

Our Tradition is to meet your needs and in so doing we invite you to tell us what your needs and interests are by completing the following:

1. I need books on the following topics:

_____

_____

_____

_____

_____

_____

2. I have the following Sams titles:

_____

_____

_____

_____

_____

_____

3. My occupation is:

_____ Scientist, Engineer
_____ Personal computerist
_____ Technician, Serviceman
_____ Educator
_____ Student
_____ D P Professional
_____ Business owner
_____ Computer store owner
_____ Home hobbyist
Other _____

_____

Name (print) _____

Address _____

City _____ State _____ Zip _____

Mail to: **Howard W. Sams & Co., Inc.**
Marketing Dept. #CBS1/80
4300 W. 62nd St., P.O. Box 7092
Indianapolis, Indiana 46206

22188

SAMS

VIC 20:

50 Easy-to-Run Computer Games

- Packed with excitement . . . good entertainment.
- Excellent variety of interesting tasks.
- Helps develop basic game programming skills.
- Easily programmed—each game is less than 30 lines, most are between 15 and 20 statements.
- Program logic is easy to follow.
- Games cover wide range of skill and ability levels.
- Challenges your dexterity . . . manual and mental.

Howard W. Sams & Co., Inc.
4300 West 62nd Street, Indianapolis, Indiana 46268 U.S.A.

$5.95/22188

ISBN: 0-672-22188-8